Sveučilište u Zagrebu
Fakultet strojarstva i brodogradnje

DIPLOMSKI RAD

Voditelj rada:
prof.dr.sc.Nedeljko Štefanić

Ivan Šimić

Zagreb, 2010

Sveučilište u Zagrebu
Fakultet strojarstva i brodogradnje

DIPLOMSKI RAD

SISTEMATIZACIJA METODA NELINEARNOG PROGRAMIRANJA

Voditelj rada

prof.dr.sc.Nedeljko Štefanić Ivan Šimić

Zagreb, 2010

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

DIPLOMSKI ZADATAK

Student: **IVAN ŠIMIĆ** Mat. br.: 0035154779

Naslov: **SISTEMATIZACIJA METODA NELINEARNOG PROGRAMIRANJA**

Opis zadatka:

U tehnološkim procesima vrlo se često pojavljuju nelinearne funkcije cilja koje je potrebno optimizirati. Kao kriterij nalaženja optimalnog rješenja javlja se minimalna hrapavost, maksimalna čvrstoća i drugi. Za iznalaženje optimalnog rješenja eksperimentom dobivene funkcije cilja na raspolaganju je veliki broj analitičkih i numeričkih metoda.

U radu je potrebno:

a) opisati matematički model nelinearnog programiranja
b) sistematizirati metode optimiranja nelinearnih funkcija cilja
c) proizvoljno odabrati jedan tehnološki proces kod kojeg se može očekivati nelinearna funkcija cilja kao rezultat provedenog eksperimenta
d) proizvoljno odabranom metodom iznaći optimalno rješenje
e) prema dobivenom rješenju izraditi podloge za odlučivanje kojima će se koristiti tehnološka priprema rada
f) navesti programske pakete koji su pogodni za pronalaženje optimuma nelinearnih funkcija.

Zadatak zadan:
11. ožujka 2010.

Rok predaje rada:
Ožujak 2011.

Zadatak zadao:
Prof.dr.sc. Nedeljko Štefanić

Predsjednik Povjerenstva:
Prof. dr. sc. Franjo Cajner

Sažetak rada

U radu su navedene i ukratko opisane sve grane operacijskih istraživanja pri čemu je poseban naglasak stavljen na nelinearno programiranje.

Opisane su metode nelinearnog programiranja uz brojne primjere rješavanja raznih nelinearnih problema. Uz pomoć metode nasumične pretrage rješeni su problemi optimizacije parametara obrade za nekoliko praktičnih tehnoloških procesa.

U završnom dijelu rada navodi se i opisuje tržišno dostupne programske pakete namjenjene rješavanju problema iz nelinearnog programiranja.

Sadržaj

Popis slika	5
Popis tablica	6
Popis oznaka i mjernih jedinica fizikalnih veličina	7
Izjava o samostalnosti pisanja rada	10
1.Opis matematičkog modela nelinearnog programiranja	11
1.1.Uvod	11
1.2. Općeniti matematički model nelinearnog programiranja	21
1.3. Princip rada klasičnih analitičkih metoda	24
1.4. Jednodimenzionalna optimizacija	30
1.5. Optimizacija NP problema sa ograničenjima	40
2. Sistematizacija metoda nelinearnog programiranja	52
2.1. Opća podjela metoda nelinearnog optimiranja	52
2.2. Podjela prema načinu rada	53
2.3. Podjela metoda optimiranja prema broju varijabli	56
3. Nelinearne funkcije cilja u tehnološkim procesima	57
3.1.Optimizacija tehnoloških parametara tokarenja za čelik 40CrMnMo	57
3.2.Izrada podloga za odlučivanje za tehnološku pripremu rada za tokarenje čelika 40CrMnMo	
3.3.Optimizacija tehnoloških parametara glodanja	62
3.4.Optimizacija tehnoloških parametara brušenja	64
4. Programski paketi pogodni za pronalaženje optimuma nelinearnih funkcija	67
4.1.Podjela prema funkcionalnosti	67
4.2.Opis nekih programskih paketa pogodnih za NP	72
5.Zaključak	74
6.Literatura	75

Popis slika

Slika 1.1. Podjela matoda Opearacijskih istraživanja 12

Slika 1.2. Podjela statističkih metoda 13

Slika 1.3. Podjela metoda matematičkog programiranja. 14

Slika 1.4. Podjela metoda stohastičkih procesa 15

Slika 1.5. Podjela suvremenih metoda optimizacije 16

Slika 1.6. Podjela optimiranja 17

Sl.1.7. Lokalni i globalni ekstremi 22

Slika 1.8. Prikaz funkcije cilja ovisne o dvije nezavisne varijable 24

Slika 1.9. Problem određivanja optimuma 25

Slika 1.10. Problem sa jednom varijablom 26

Slika 1.11. Iterativni proces optimizacije jednodimenzijskih numeričkih metoda 31

Slika 1.12. Sistematizacija numeričkih jednodimenzionalnih metoda za NP 32

Slika 1.13. Unimodalne funkcije 33

Slika 1.14. Ishod dvaju pokusa za slučajeve (a) $f_1 < f_2$; (b) $f_1 > f_2$ i (c) $f_1 = f_2$ 34 34

Slika 1.15. Unimodalna funkcija $f(x) = x^2-x+50$ 36

Slika 1.16. Načelo rada metode iscrpne pretrage 38

Slika 1.17. Izravne metode za probleme NP sa ograničenjima 41

Slika 1.18. Neizravne metode za probleme NP sa ograničenjima 51

Slika 2.1. Podjela NP metoda prema načinu rada 53

Slika 2.2. Sistematizacija metoda nelinearnog statičkog programiranja 55

Slika 3.1. Ovisnost funkcije cilja o radijusu alata uz optimalne parametre 61

Popis tablica

Tablica 1.1. Proračun metodom neograničene pretrage sa stalnim korakom 36

Tablica 1.2. Proračun metodom neograničene pretrage sa stalnim korakom 37

Tablica 1.3. Iznosa intervala nesigurnosti za svaki od ukupno n postupaka 39

Tablica 1.4.Primjer proračuna metodom iscrpne pretrage 39

Tablica 1.5. Primjer proračuna metodom nasumične pretrage 43

Tablica 1.6. Raspon vrijednosti varijabli i preciznost 45

Tablica 1.7. Generiranje slučajnih brojeva i iznosa varijabli 45

Tablica 1.8. Generiranje slučajnih brojeva i iznosa varijabli 46

Tablica 1.9. Tablica prvog riješenja 46

Tablica 1.10. Različiti rasponi za varijable 48

Tablica 2.1. Podjela metoda nelinearnog programiranja prema broju varijabli 56

Tablica 3.1. Generiranje vrijednosti varijabli ovisno o vrijednosti sl. broja 59

Tablica 3.2. Pravilo novog generiranja vrijednosti varijabli 59

Tablica 3.3. Tablica 3.3. Pravila posljednjeg generiranja vrijednosti varijabli 60

Tablica 3.4. Ovisnost funkcije cilja o radijusu alata uz optimalne parametre 61

Tablica 3.5. Pravilo prvog generiranja varijabli (glodanje) 63

Tablica 3.6. Pravilo drugog generiranja varijabli (glodanje) 63

Tablica 3.7. Pravilo prvog generiranja varijabli (brušenje) 65

Tablica 4.1.Osnove numeričke matematike korištene u programskim paketima 70

Popis oznaka i mjernih jedinica fizikalnih veličina

(prema redosljedu pojavljivanja)

$F(X)$	Funkcija cilja ovisna o vektoru varijabli X
X	Vektor varijabli
Rn	Euklidski prostor
S	Skup ograničenja
R	Kodomena euklidskog prostora
X^T	Transponirani vektor varijabli
$h_i(X)$	Ograničenja zadana u obliku jednadžbi
$g_j(X)$	Ograničenja zadana u obliku jednadžbi
G	Matematički skup pozitivnih brojeva
$f(x)$	Funkcija cilja ovisna o jednoj varijabli
$f(x_o)$	Vrijednost funkcije u točki x_0
$f'(x_o)$	Prva derivacija funkcije u točki x_0
$f''(x_o)$	Druga derivacija funkcije u točki x_0
∇f	Gradijent vektor
H_f	Hesseova matrica
S_i	Smjer kretanja pretraživanja kod jednodimenzionalnih numeričkih metoda
λ_i^*	Duljina koraka kretanja pretraživanja kod jednodimenzionalnih numeričkih metoda
s	Korak pretrage u metodi eliminacije
$f_{(xj)}$	Vrijednost funkcije cilja u točki s desne strane od ishodišta

$f\,(x_{\text{-j}})$	Vrijednost funkcije cilja u točki s ljeve strane od ishodišta
x_s	Početna točka kod metode iscrpne pretrage
x_f	Krajnja točka kod metode iscrpne pretrage
L_0	Početni interval nesigurnosti
L_n	Krajnji interval nesigurnosti
P	Dio ukupnog intervala nesigurnosti gdje se nalazi optimum kod metode iscrpne pretrage
$sl.br._1$	Slučajni broj 1
$sl.br._2$	Slučajni broj 2
k	Korak za metodu nasumične pretrage
Fc_{opt}	Optimalan iznos funkcije cilja
R_p	Raspon za metodu nasumične pretrage
F_{max}	Masimalni iznos funkcije cilja
k_v	Broj vrhova poliedra kod simplex metode
n_v	Broj varijabli
$r_{i,j}$	slučajni broj na intervalu od 0 do 1
X_0	Centroid vrhova u metodi u complex metodi
X_h	Točka u kojoj je trnutno vrijednost funkcije cilja najveća u complex metodi
X_k	Točka koja se pronalazi zrcaljenjem u complex metodi
$F(X_h)$	vrijednost funkcije cilja u točki X_h
$F(X_r)$	vrijednost funkcije cilja u točki X_r
α	Koeficijent pretrage u complex metodi

ε_1		Mimimalni iznos koeficijenta pretrage
ε_2		Mimimalni iznos complex-a
ε_3		Mimimalni iznos standardne devijacije vrijednosti funkcije
a_p	(mm)	Dubina rezanja
v_c	(m/min)	Brzina rezanja
f	(mm)	posmak
r_e	(mm)	radijus zaobljenja tokarskog noža
f_{min}	(mm)	minimalni dozvoljeni posmak
f_{max}	(mm)	maksimalni dozvoljeni posmak
a_{pmin}	(mm)	Minimalna dubina rezanja
a_{pmax}	(mm)	Maksimalna dubina rezanja
R_a	(μm)	Prosječno odstupanje profila
k_{var}		korak kod metode nasumične pretrage
R_{var}		Raspon kod metode nasumične pretrage
φ		Kut namještanja alata

Izjava o samostalnosti pisanja rada

Izjavljujem da sam rad napisao samostalno i bez ičije pomoći.

Ivan Šimić

1.Opis matematičkog modela nelinearnog programiranja

1.1.Uvod

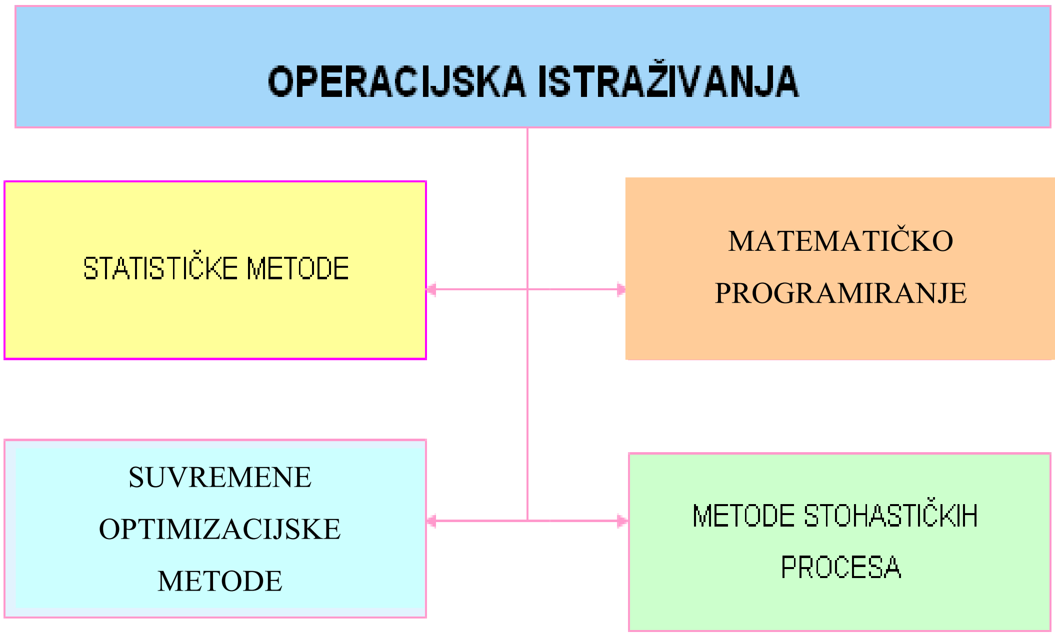
Područje „Operacijska istraživanja“ (IO) javlja se početkom drugog svjetskog rata te se zasniva na proučavanju svih fenomena koji se pojavljuju usljed vojnih operacija i fenomena koji utječu na iste. Potrebno je bilo usukladiti na najbolji način ekonomske, proizvodne, društvene i vojne čimbenike kako bi se postigla maksimalna učiinkovitost zaraćenih država.
Operacijsko istraživanje je znanstvena disciplina koja primjenjuje kvantitativne i kvalitativne metode za proučavanje problema koji se javljaju prilikom provođenja neke operacije (ili nekog procesa) i pronalazi najbolje moguće rješenje (optimum) za provođenje te operacije.

Svaka realna operacija je cjelina koja se sastoji od velikog broja podoperacija odnosno segmenata stoga je bitno naglasiti da pronalaženje optimuma svih segmenata nije nužno i optimum cjeline (operacije).

Da bi operacija bila predmet proučavanja OI ona treba imati sljedeće značajke:

- Jasno određenje cilja
- Postojanje brojnih i različitih rješenja koja je nemoguće sagledati ili procjeniti i donjeti optimalno rješenje intuicijom ili iskustvom istraživača
- Mogućnost kvantifikacije odnosno primjene matematičkih modela i metoda.

Na slici 1.1. prikazana je podjela metoda operacijskih istraživanja na četiri glavna područja odnosno na statističke metode, matematičko programiranje, suvremene metode te metode stohastičkih procesa.



Slika 1.1.Podjela metoda opearacijskih istraživanja

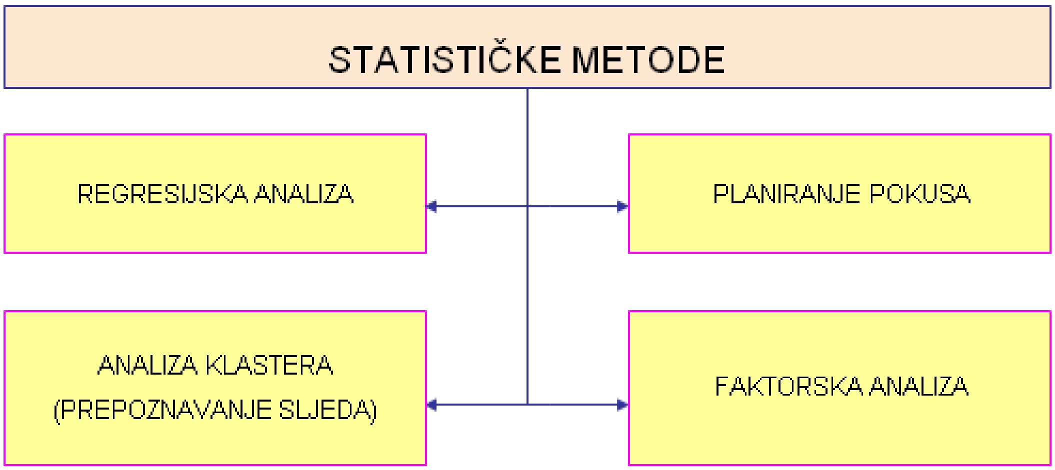
Statističke metode omogućuju analizu podataka nakon vršenja pokusa i stvaranje empirijskih modela kako bi najvjernije opisale realno stanje promatranog sustava, na sljedećoj slici prikazana je podjela statističkih metoda na regresijsku analizu, planiranje pokusa,analizu klastera te na faktorsku analizu.

Planiranje pokusa (eng. „Design of experiments“) koristi se za modeliranje funkcije cilja koja vjerno opisuje neku stvarnu fizikalnu veličinu na koju utječu određene varijable.

Podaci o iznosima varijabli dobivaju se mjerenjem prilikom provođenja pokusa nakon čega se odgovarajućom obradom tih podataka dobiva funkcija cilja koju je potrebno optimirati.

Faktorska analiza koristi se u slučaju kada se pretpostavlja da na funkciju cilja (odnosno na neku ekonomsku ili fizičku veličinu) utječe određeni broj faktora odnosno varijabli premda to nije dokazano. Stoga je potrebno odreditikoje varijable utječu i koliko je značajan njihov utjecaj na funkciju cilja.

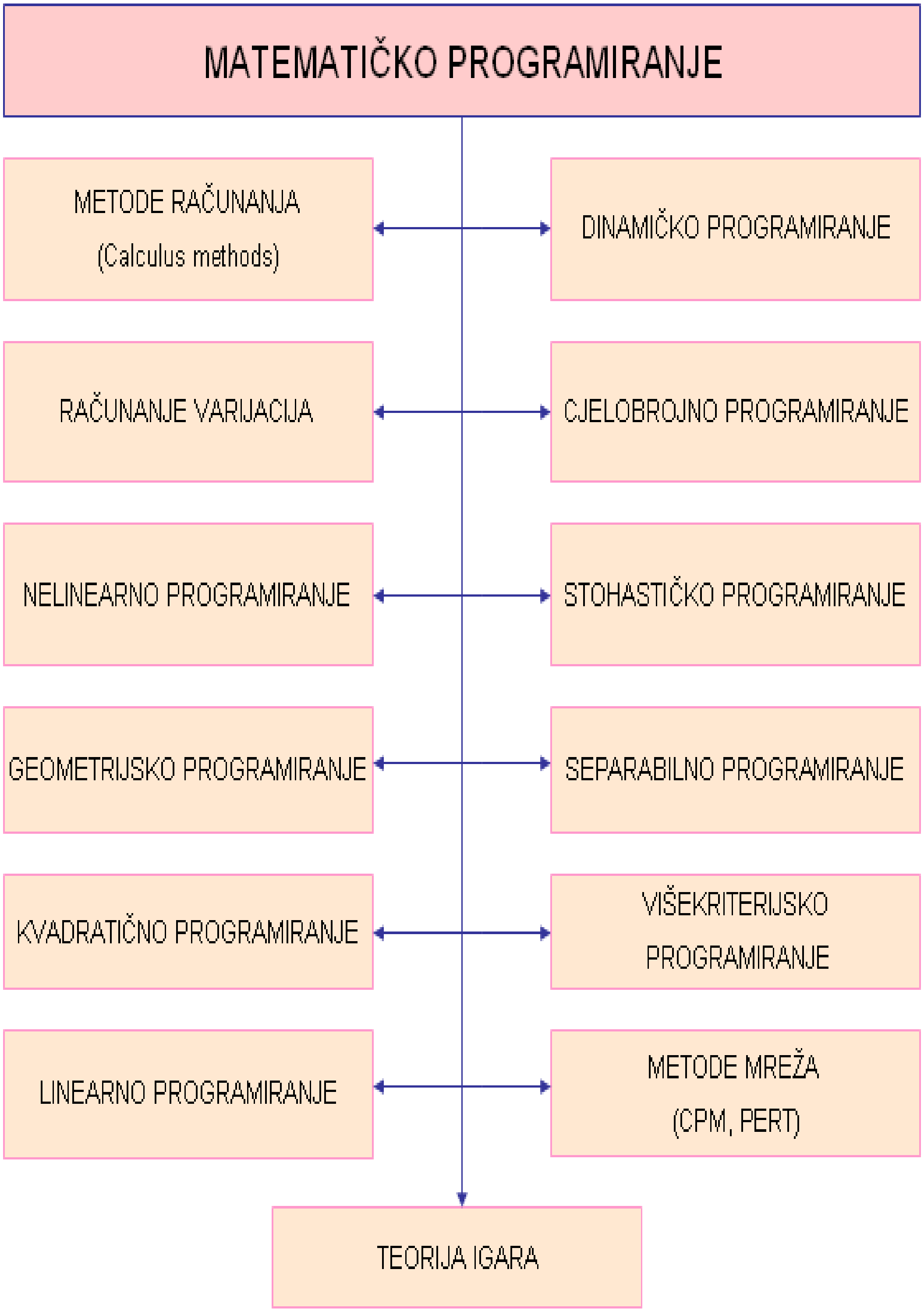
Na slici 1.2. prikazana je podjela statističkih metoda na četiri glavne skupine i to na regresijsku analizu, planiranje pokusa, analizu klastera i faktorsku analizu.



Slika 1.2. Podjela statističkih metoda

Na slici 1.3. prikazana je sistematizacija metoda matematičkog programiranja u koje spada i nelinearno programiranje.

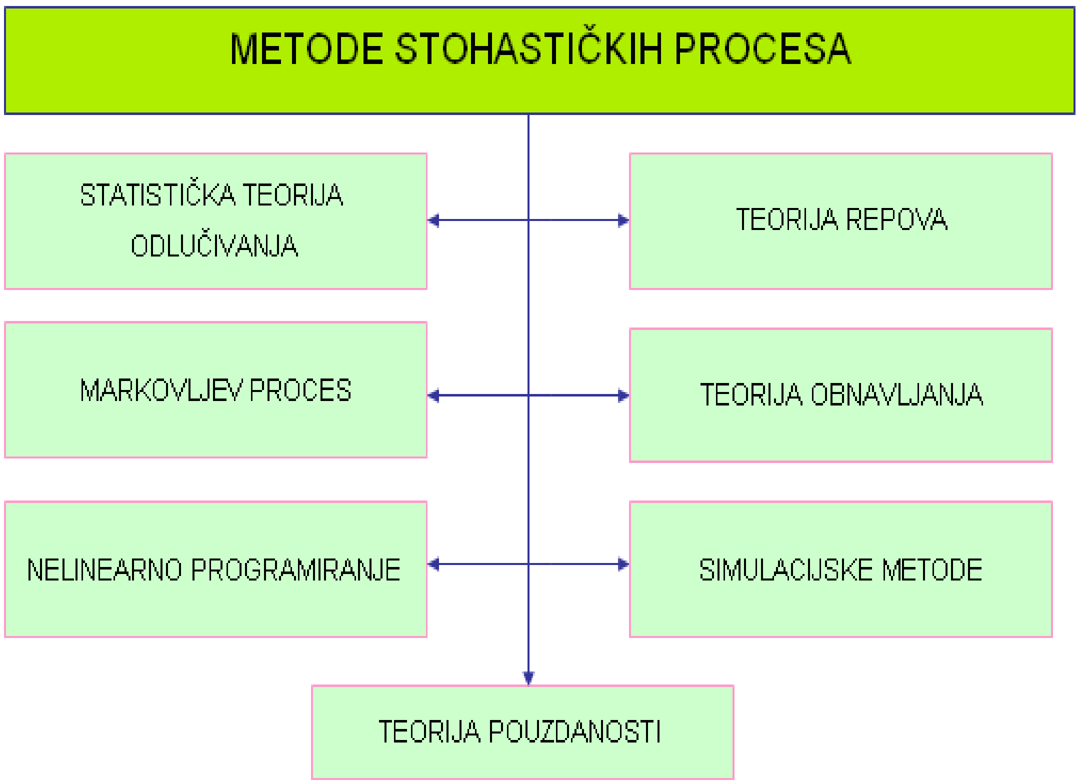
Matematičko programiranje koje se naziva još i optimizacijske metode dijeli se na metode računanja, dinamičko programiranje, računanje varijacija, cjelobrojno programiranje, nelinearno programiranje, stohastičko programiranje, geometrijsko programiranje, separabilno programiranje, kvadratično programiranje, višekriterijsko programiranje, linearno programiranje, metode mreža i teoriju igara.



Slika 1.3. Podjela metoda matematičkog programiranja

Metode stohastičkih procesa koriste se za analizu problema opisanih skupom varijabli koje se podvrgavaju poznatim razdiobama vjerojatnosti.

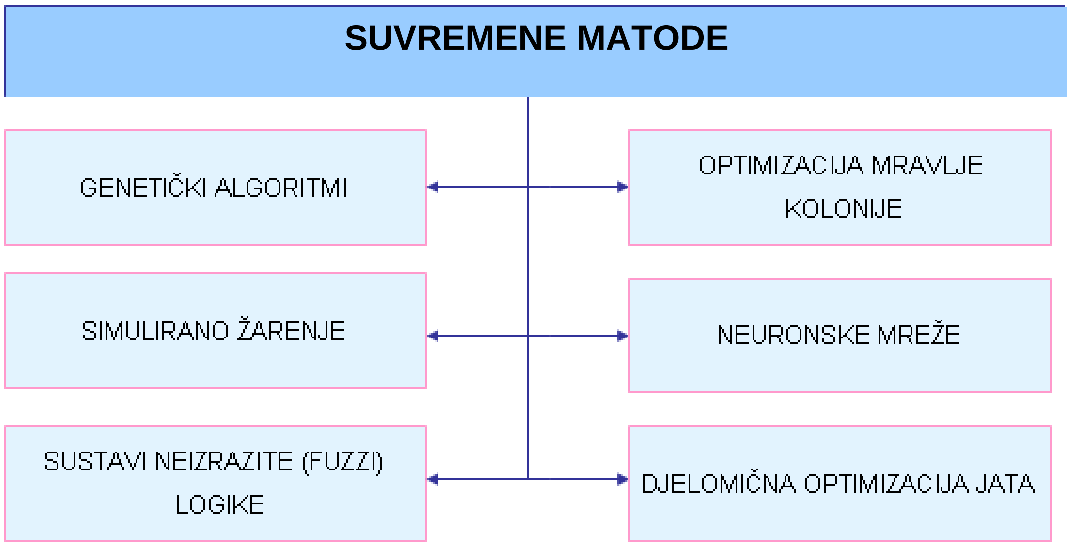
Metode stohastičkih procesa dijele se na statističku teoriju odlučivanja, teoriju repova, Markovljev proces, teoriju obnavljanja, nelinearno programiranje, simulacijske metode i teoriju pouzdanosti.



Slika 1.4. Podjela metoda stohastičkih procesa

Za uspješnu primjenu stohastičkih metoda potrebno je određeno vrijeme promatrati neki proces kako bi se prikupili podaci na temelju kojih se donosi zaključak prema kojoj se statističkoj razdiobi podaci podvrgavaju stoga ove metode nisu pogodne za optimizaciju potpuno novih procesa.

Suvremene metode dijele se na genetičke algoritme, optimizaciju mravlje kolonije, simulirano žarenje, neuronske mreže, sustave neizrazite logike te na djelomičnu optimizaciju jata te zauzimaju veoma važno mjesto u operacijskim istraživanjima kao i u industriji općenito.

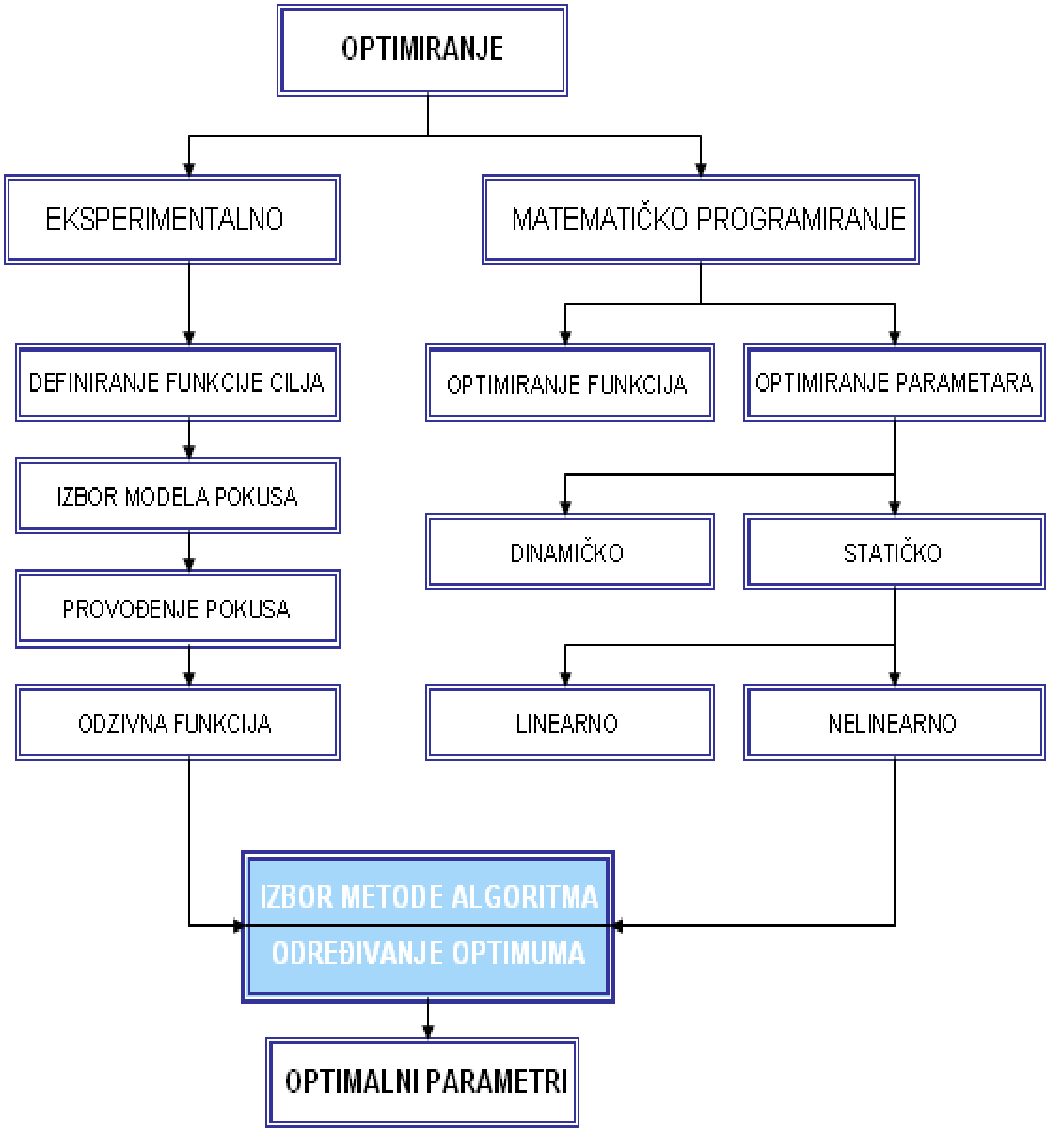


Slika 1.5. Podjela suvremenih metoda optimizacije

Optimiranje

Zadatak optimiranja je iznaći najbolje moguće rješenje (najbolju vrijednost funkcije cilja) od više mogućih rješenja odnosno odrediti optimalne vrijednosti varijabli koji utječu na vrijednost funkcije cilja.

Na slici 1.6. prikazana je jedna od mogućih podjela optimiranja prema [3] .



Slika 1.6.Podjela optimiranja

Eksperimentalno optimiranje spada u statističke metode operacionih istraživanja.

U praksi postoji veliki broj slučajeva u kojima nije moguće postaviti dovoljno dobar matematički model zbog prevelike kompleksnosti ili zbog nepostojanja fizičkih modela pomoću kojih bismo opisali stvarni sustav. U tim slučajevima neophodno je primjeniti eksperimentalno optimiranje odnosno statističke metode.

Zadaća provođenja eksperimenata jest utvrditi matematičku vezu između funkcije cilja i parametara koja se naziva *odzivna funkcija.* Nakon određivanja odzivne funkcije relativno je lako odrediti optimalan iznos parametara.

Za sve druge vrste problema koje je moguće dovoljno točno matematički opisati odnosno postaviti matematički model primjenjuje se matematičko optimiranje koje se dijeli na optimiranje funkcija i optimiranje parametara.

Optimiranje funkcija bavi se traženjem minimalnog ili maksimalnog iznosa funkcije neovisno o ograničenjima dok se optimiranje parametara kao što i samo ime kaže bavi određivanjem optimalnog iznosa parametara koji utječu na funkciju cilja.

Programiranje parametara može biti dinamičko i statično.

U statičnom programiranju utrošak resursa po parametru (varijabli) u funkcijama ograničenja je stalan u dok u dinamičkom to nije nužno.

Prema obliku matematičkih funkcija ograničenja i funkcije cilja statičko optimiranje parametara dijeli se dalje na linearno i nelinearno optimiranje parametara.

Linearno programiranje je primjenjivo na veliki broj stvarnih problema u prehrani, u poljoprivrednoj i ostaloj proizvodnji i sl. te se može riješavati grafički, analitički i simpleks metodom. Međutim postoje brojni slučajevi u praksi u kojima veze među parametrima u funkciji cilja i/ili funkcijama ograničenja nisu linearne tako da ih je nemoguće riješiti primjenom klasične simpleks metode stoga je bilo potrebno razviti posebnu granu optimiranja koje se naziva nelinearno optimiranje (nelinearno programiranje-NP).

U svakodnevnom životu potrebno je ostvariti brojne ciljeve na čije ostvarivanje utječu brojna ograničenja (bez ograničenja raspolagali bismo sa neograničenom količinom resursa stoga ne bi bilo potrebno optimirati jer bi svaki cilj bilo moguće ostvariti).

Ciljevi koji se postavljaju najčešće su maksimiziranje zarade, proizvodne količine i iskorištenja kapaciteta te minimalnizacija troškova, trošenja resursa, vremena proizvodnje i transporta i sl .

Funkcije cilja ovise o raznim varijablama za koje je karakteristično da svaka varijabla zahtijeva određenu količinu resursa.

Količina resursa kojim poduzeća, odjeli ili pojedinci raspolažu predstavljaju ograničenja.
Ograničenja su površina zemljišta, dubina/širina plovne rijeke i slična ograničenja u prirodi, novčana sredstva, vrijeme (rokovi), potražnja za proizvodima , nosivost i broj transportnih sredstava, ograničenja prostora (površinska ili volumenska ograničenja), broj radnika i opreme, količina repromaterijala, proizvoda i poluproizvoda na skladištu, dostupnost sirovina, vode i energije i ostala ograničenja.

Da bi se ostvarilo uspješno rukovođenje poslovnih, proizvodnih i ostalih procesa potrebno je upotrijebiti minimalnu količinu resursa za postizanje željenih ciljeva.

Kako bi se matematičke metode mogle primjeniti kao alat za izračunavanje optimalnog iznosa varijabli potrebnih za postizanje cilja nužan je matematički opis ovisnosti funkcije cilja (i funkcija ograničenja) o varijablama .

Kada je matematički model postavljen tada je relativno jednostavno doći do rješenja upotrebom jednog od brojnih programskih paketa za rješavanje NP problema.
Načelno svi problemi mogu se riješiti bez računala upotrebom metoda za NP, no složenije probleme NLP-a neophodno je rješavati na računalu primjenom odgovarajućih programskih paketa zbog uštede vremena i smanjenja mogućnosti pogreške(današnji software-i mogu rješavati probleme sa nekoliko desetaka tisuća do nekoliko milijuna ograničenja i varijabli što je gotovo neizvediv posao za čovijeka).
Uz to bitno je spomenuti da večina programskih paketa za NP automatski izabire jednu od više metoda koja je najpovoljnija za rješavanje zadanog problema.

Pri praktičnoj primjeni nelinearnog programiranja potrebno je provesti sljedeće korake:

Općenito nelinearno programiranje (NP) je skup matematičkih metoda koje se bave problemom optimizacije sustava unutar zadanih ograničenja.

U realnim problemima NP-a mogu se javiti tri slučaja:

- Linearna funkcija cilja sa nelinearnim ograničenjima
- Nelinearna funkcija cilja sa linearnim ograničenjima
- Nelinearna funkcija cilja sa nelinearnim ograničenjima

1.2. Općeniti matematički model nelinearnog programiranja

Zadana je nelinearna funkcija F(X) kojoj je potrebno pronaći ekstrem (minimum ili maksimum).

$F(X)=F(x_1,x_2,x_3.....x_n)\rightarrow$max/min

$F(X):S\rightarrow R$

Funkcija $F(X)$ definirana je na skupu S iz Euklidskog prostora R^n, pri čemu ona preslikava domenu S u kodomenu R.

Skup S je podskup skupa R^n .

Zadatak analitičkih metoda nelinearnog programiranja je pronaći transponirani vektor varijabli X^T koji će omogućiti najmanju ili najveću vrijednost funkcije cilja *F(X)* uz poštivanje zadanih ograničenja u skupu ograničenja S.

$$X=[x_1,x_2,x_3.....x_n]$$

$$X^T=\begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix}$$

Ograničenja mogu biti zadana u obliku jednadžbi $h_i(X)$, i=1,2,3…m, u obliku nejednadžbi $g_j(X)$, j=1,2,3…r i u obliku ograničenja nenegativnosti X ε G odnosno $X_i \geq 0$.

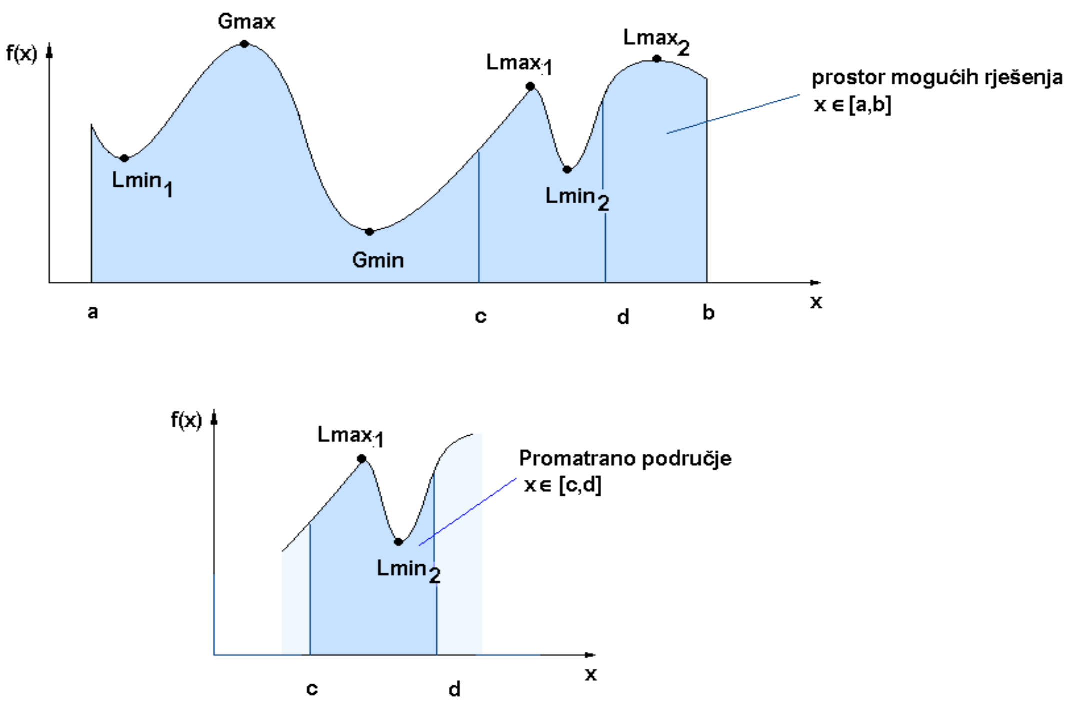
$$\left.\begin{matrix} h_i(X) \\ g_j(X) \\ x \in G \end{matrix}\right\} \in S \Big|$$

Većina metoda radi na nčin da traži minimum funkcije cilja $F(X)$ .U slučajevima kada treba odrediti maksimum $F(X)$ potrebno je tražiti minimalno rješenje funkcije $-F(X)$.

1.2.1. Globalni i lokalni ekstremi u problemima sa jednom varijablom

Ekstremi neke funkcije cilja dijele se na lokalne i globalne ekstreme. Globalni ekstremi su točke u kojima funkcija cilja poprima svoju najveću ili najmanju vrijednost u prostoru mogućih riješenja.

Lokalni ekstremi su točke u kojima funkcija poprima najmanju ili najveću vrijednost na nekom promatranom prostoru (lokalitetu) koji je manji od cjelokupnog prostora mogućih rješenja.



Sl.1.7. Lokalni i globalni ekstremi

Na slici 1.7. vidi se da na prostoru gdje je $x \in [c,d]$ (koji je samo mali dio ukupnog prostora mogućih rješenja $x \in [a,b]$) postoji jedan minimum i maksimum.
Na promatranom području, funkcija u te dvije točke (L_{max1} i L_{min2}) ima svoju minimalnu i maksimalnu vrijednost, no kada se promatra cijeli prostor mogućih rješenja postoje druge točke u kojima funkcija cilja poprima još veću (G_{max}) i još nižu vrijednost (G_{min}) stoga se nazivaju globalnim ekstremima.

Ako je:

- $f'(x_o)=0$ i $f''(x_o)>0$, tada $f(x)$ ima minimum u točki x_o
- $f'(x_o)=0$ i $f''(x_o)<0$, tada $f(x)$ ima maksimum u točki x_o

Da bi neka točka bila ekstrem prva derivacija funkcije $f(x)$ u toj točki mora biti jednaka nuli, dok nam predznak druge derivacije u toj točki govori o kakvoj vrsti ekstrema se radi.

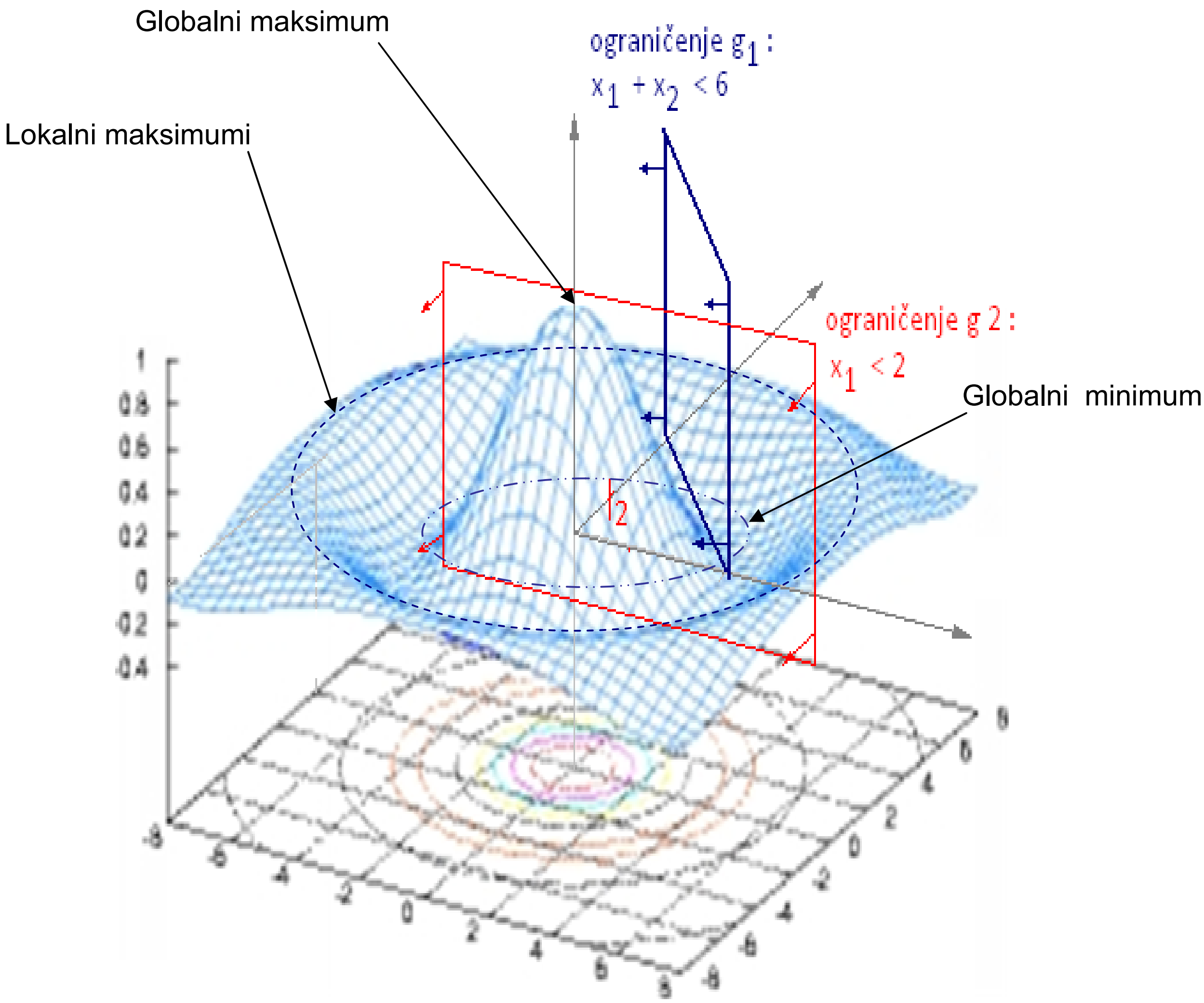
Ako vrijedi :

- $f(x)<f(x_o)$ na intervalu $x \in (a,b)$ tada je točka x_o globalni maksimum
- $f(x)>f(x_o)$ na intervalu $x \in (a,b)$ tada je točka x_o globalni minimum

U problemima sa više varijabli sve gore navedene zakonitosti također vrijede .

1.2.2.. Globalni i lokalni ekstremi u problemima sa više varijabli

Na sljedećim slikama prikazani su neki zanimljivi problemi kako bi se ukazalo na svu problematiku koja se javlja u nelinearnom programiranju u multivarijabilnim problemima.

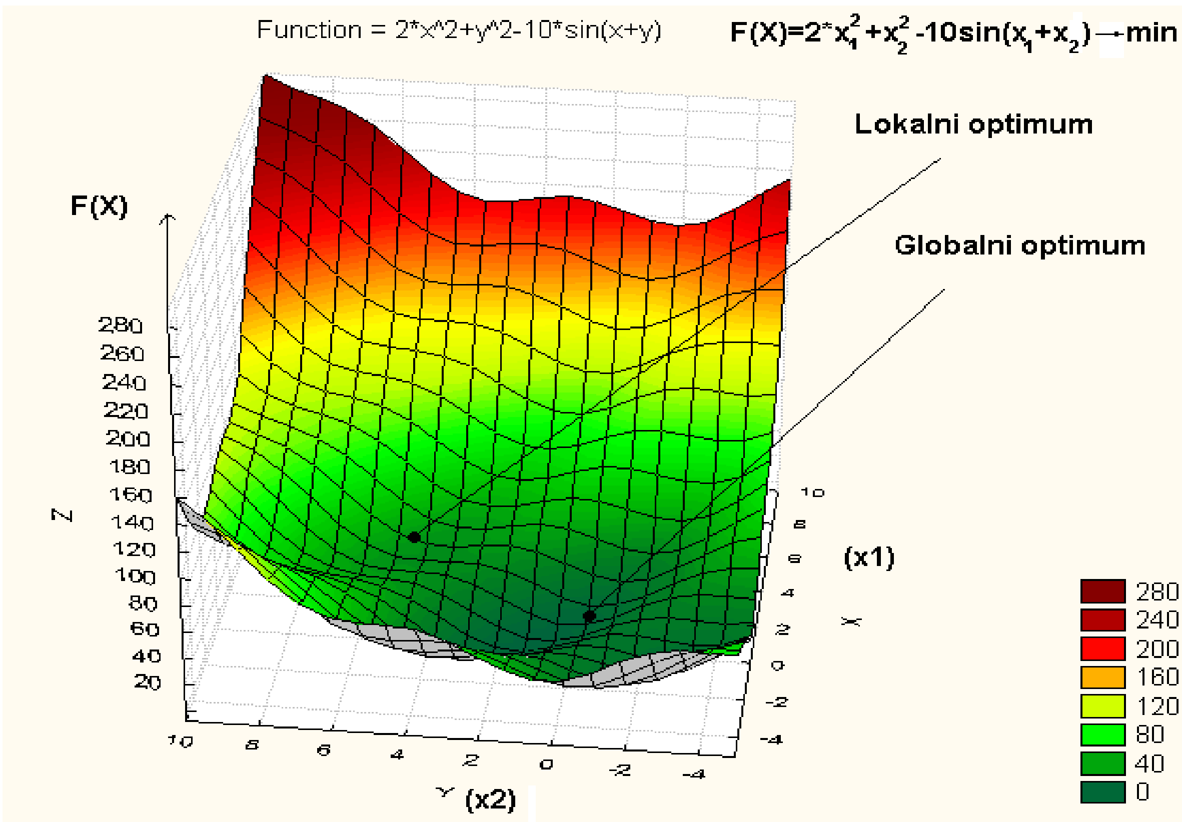


Slika 1.8. Prikaz funkcije cilja ovisne o dvije nezavisne varijable

Na slici 1.8. prikazana je funkcija cilja ovisna o dvije varijable x_1 i x_2 koja sadrži jedan globalni maksimum i beskonačno mnogo lokalnih minimuma i maksimuma.

Isprekidana gornja krivulja označava beskonačno mnogo lokalnih maksimuma dok donja isprekidana kružnica označava beskonačno mnogo globalnih minimuma.

Ako se traži minimum sasvim je nebitno koje od globalnih minimuma izaberemo jer svi uzrokuju istu minimalnu vrijednost funkcije cilja. Ukoliko je potrebno odrediti maksimum tada je samo jedno rješenje koje zadovoljava.



Slika 1.9. Problem određivanja optimuma

Na slici 1.9. prikazana je funkcija $F(X)=2x_1^2+x_2^2-10\cdot\sin(x_1+x_2)\rightarrow\min$.

Prikazane su dvije kritične točke od kojih je jedna lokalni ekstrem (točka T_1) a druga globalni ekstrem (točka T_2).

Grafički je moguće prikazati samo funkcije cilja ovisne samo o jednoj ili o dvije varijable.

U problemima sa više od dvije varijable problem je isti, a to je postojanje više lokalnih optimuma kod nekih funkcija cilja .

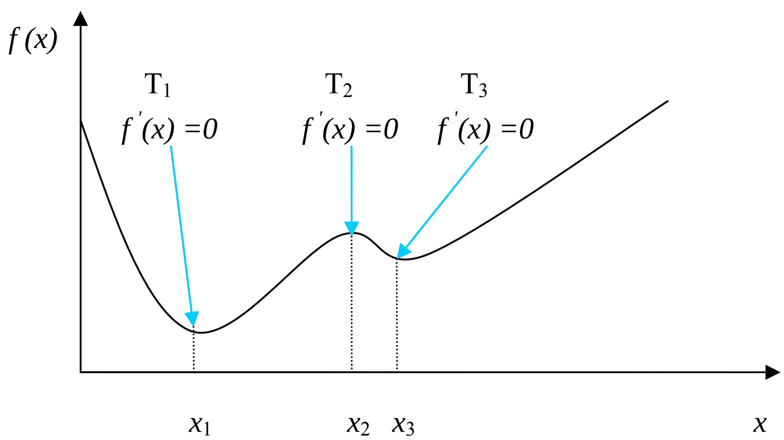
1.3. Princip rada klasičnih analitičkih metoda

Kod funkcija sa jednom kao i kod funkcija sa više varijabli potrebno je odrediti prvo stacionarne točke u kojima je prva derivacija funkcije cilja jednaka nuli. Nakon određivanja stacionarnih točaka ispituje se narav ekstrema odnosno radi li se o minimumu ili o maksimumu. Ukoliko postoji više lokalnih rješenja, ona se uspoređuju međusobno i izabire se ono koje je najbolje.

Problemi sa jednom varijablom

Neka je zadana funkcija cilja f(x) prikazana na slici 1.10. kojoj tražimo minimalni iznos.

Prvo se pronalaze točke u kojima je prva derivacija funkcije cilja jednaka nuli rješavajući jednadžbu: $\frac{df(x)}{dx} = 0$



Slika 1.10. Problem sa jednom varijablom

Rezultat rješavanja jednadžbe su tri točke T_1, T_2 i T_3 za koje se provjerava jesu li one minimumi ili maksimumi. To se provodi na način da se izračuna predznak druge derivacije u tim točkama : $\left.\frac{df^2(x)}{dx^2}\right|_T$

Ako je predznak negativan tada se radi o maksimumu a u suprotnom ako je pozitivan radi se o minimumu. U ovom slučaju postoje dva minimuma odnosno točke T_1 i T_3. Kako bi se utvrdilo koja od njih je stvarno rješenje odnosno globalni minimum uspoređuju se iznosi funcije *f* (*x*) u tim točkama. Ako je $f(x_1)< f(x_3)$ tada je točka T_3 globalni optimum, u suprotnom ako je $f(x_1)> f(x_3)$ tada je točka T1 globalni optimum. U slučaju da je $f(x_1)=f(x_3)$ obadvije točke su globalni optimumi i sasvim je svejedno koja se izabire.

U ovom slučaju točka T1 je globalni minimum.

Multivarijabilni problemi

Kod funkcija sa više varijabli za ispitivanje stacionarnih točaka gradijentne metode koriste gradijent vektor.

$$\nabla F \equiv \left[\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \dots \frac{\partial F}{\partial x_n} \right]^T$$

Gradijent vektor ∇F , odnosno gradijent je vektorsko polje koje pokazuje smjer najvećeg porasta skalarnog polja $F(X)= F(x_1 , x_2 \dots x_n)$.

Stacionarna točka je ona točka u kojoj je iznos gradijent vektora jednak nuli i ona je ujedno i ekstrem funkcije *F(X).*

$$\nabla F(\bar{X}) = \nabla F \big|_{\bar{X}} = 0$$

Stacionarne točke nazivaju se kritičkim točkama.

Nakon što je određena stacionarna točka, potrebno je odrediti prirodu ekstrema (radi li se o minimumu ili o maksimumu).

Za utvrđivanje prirode ekstrema koristi se Hesseova matrica.

Da bi se mogla primjeniti Hesseova matrica stacionarna točka koja se ispituje ne smije biti degenerativna jer je u degenerativnoj točki determinanta Hesseove matrice jednaka nuli stoga nije moguće zaključiti o kakvom se ekstremu radi.

Hesseova matrica je kvadratna matrica derivacija drugog reda funkcije *F(X)* koja opisuje lokalnu zakrivljenost viševarijabilnih funkcija, a označava se oznakom H_f

$$H_f \equiv \left[\frac{\partial^2 F}{\partial x_i \partial x_j} \right], \qquad (i, j ....., \quad n)$$

Kada je Hesseova matrica pozitivno definitna u kritičnoj točki, tada je ta točka lokalni minimum. Ako je pak Hesseova matrica u kritičnoj točki negativno definitna tada se radi o lokalnom maksimumu.

Kako bi se odredilo radi li se o pozitivnoj ili negativnoj definitnosti matrice koristi se determinanta simetrične (n x n) matrice $B \equiv [b_{ij}]$:

$$B_1 \equiv +|b_{11}| \qquad B_2 \equiv -\begin{vmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{vmatrix} \quad B_3 \equiv \begin{vmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{vmatrix}$$

$B_n \equiv (-1)^{n-1} detB$

Gdje je oblik matrica B_2 i B_3 u proširenom obliku :

$B_2 = -(b_{11} \cdot b_{22} - b_{12} \cdot b_{21})$

$$B_3 = b_{11} \cdot \begin{vmatrix} b_{22} & b_{23} \\ b_{32} & b_{33} \end{vmatrix} - b_{12} \cdot \begin{vmatrix} b_{21} & b_{23} \\ b_{31} & b_{33} \end{vmatrix} + b_{13} \cdot \begin{vmatrix} b_{21} & b_{22} \\ b_{31} & b_{32} \end{vmatrix}$$

Ako su :

- $B_1, B_2, B_3 \ldots B_n < 0$ tada je matrica B u točki $\bar{X}$ negativno definitna
- $B_1, B_2, B_3 \ldots B_n > 0$ tada je matrica B u točki $\bar{X}$ pozitivno definitna

- $B_1, B_2, B_3 \ldots B_p < 0$ ($p<n$) a ostali B_p do B_n su nule tad je matrica B u točki $\bar{X}$ negativno semidefinitna
- $B_1, B_2, B_3 \ldots B_p > 0$ ($p<n$) a ostali B_p do B_n su nule tad je matrica B u točki $\bar{X}$ pozitivno semidefinitna

Pri određivanju prirode ekstrema, ako je:

- $\nabla F(\bar{X}) = 0$ i ako je $H_f \big|_{\bar{X}}$ negativno definitna tada funkcija $F(\vec{X})$ ima minimum u točki $\bar{X}$
- $\nabla F(\bar{X}) = 0$ i ako je $H_f \big|_{\bar{X}}$ pozitivno definitna tada funkcija $F(\vec{X})$ ima maksimum u točki $\bar{X}$

Stoga ako vrijedi :

- $F(\vec{X}) > F(\bar{X})$ na cijelom prostoru mogućih rješenja tada je točka $\bar{X}$ globalni minimum
- $F(\vec{X}) < F(\bar{X})$ na cijelom prostoru mogućih rješenja tada je točka $\bar{X}$ globalni maksimum

1.4. Jednodimenzionalna optimizacija

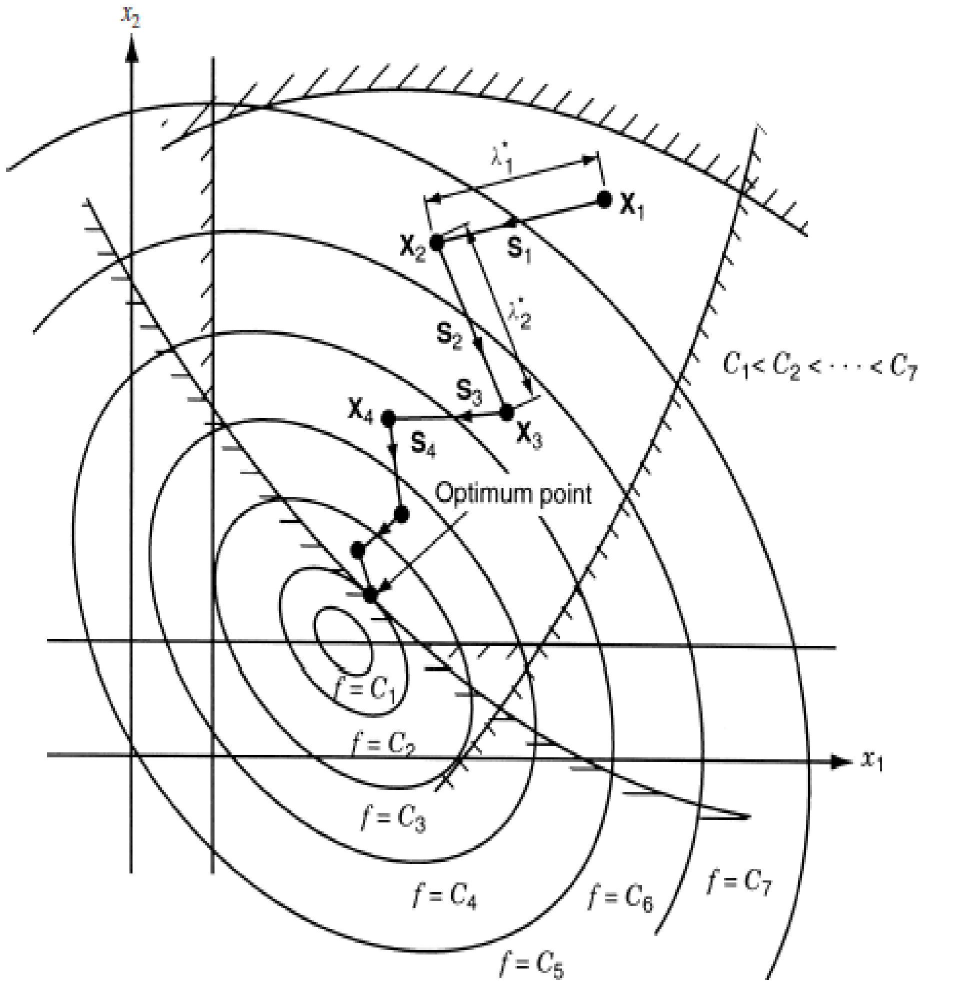
Neke probleme nije moguće rješiti analitičkim metodama,to se događa u sljedećim slučajevima:

- Funkcije cilj i/ili ograničenja su presložene
- Funkcija cilja je diskontinuirana
- Funkcija cilja je dvostruko nederivabilna stoga nije moguće utvrditi radi li se o minimumu ili o maksimumu
- Funkcija cilja nije izražena kao eksplicitna funkcija varijabli

U tim slučajevima koriste se numeričke metode koje rade na drugačiji način od klasičnih analitičkih metoda računanja derivacija. Kod analitičkih metoda računanje vrijednosti funkcije cilja nakon određivanja vrijednosti varijabli odvija se kao zadnji korak procesa dok je kod numeričkih metoda obrnuto. Numeričke metode traže prvo vrijednost funkcije cilja računajući njen iznos za razne kombinacije varijabli nakon čega se uspoređuju sva rješenja i odabire optimalno (nakon čega se usvajaju pripadajući iznosi varijabli optimalnog rješenja).

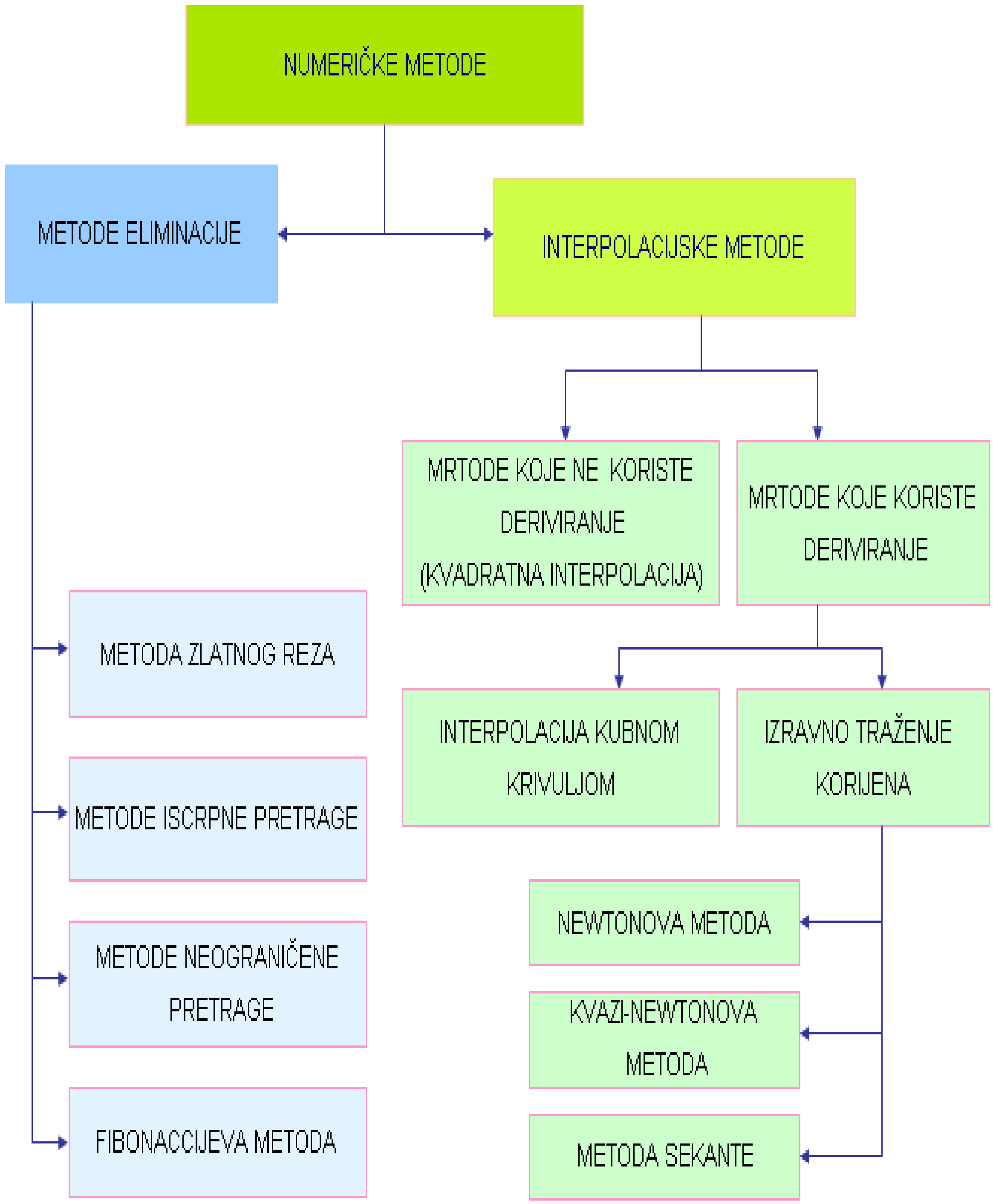
Princip rada jednodimenzionalnih numeričkih metoda:

1. Početi pretragu u početnoj točki $X_{1.}$
2. Pronaći odgovarajući smjer kretanja pretraživanja S_i (za početak i =1) koji generalno pokazuje u kojem je smjeru optimum.
3. Pronaći odgovarajuću duljinu koraka kretanja λ_i^* u smjeru S_i kako bi se odredila nova točka u kojoj se provodi aproksimacija.
4. Provođenje nove aproksimacije u točki X_{i+1}
   $X_{i+1} = X_i + \lambda_i^{*\cdot} S_i$
5. Provjeriti da li funkcija u točki X_{i+1} ima svoj optimum. Ako je u X_{i+1} optimum tada zaustaviti postupak, ako nije prelazi se na ponovno provođenje drugog koraka uz to da je sad $i = i+1$.



Slika 1.11. Iterativni proces optimizacije jednodimenzijskih numeričkih metoda

Na slici 1.11. prema [1] prikazan je iterativni proces optimizacije numeričkih metoda za hipotetski slučaj problema sa dvije varijable. Iz formule $X_{i+1} = Xi + \lambda_i^{*.} S_i$ vidi se da na učinkovitost ovih optimizacijskih metoda ovisi o učinkovitosti pronalaženja vrijednosti $\lambda_i^{*} i\, S_i$.
Kada treba pronaći minimum funkcije cilja $f(X)$, problem pronalaženja vrijednosti λ_i^* svodi se na određivanje vrijednosti $\lambda_i = \lambda_i^*$ koja minimizira $f\,(X+1) = f\,(Xi + \lambda_i^{\cdot}\, S_i) = f(\lambda_i^*)$ uz fiksne vrijednosti X_i $i\ S_i$. Na taj način funkcija f ovisi samo o jednoj dimenziji λ_i . Zbog navedenog razloga metode koje traže λ_i^* nazivaju se *jednodimenzijske metode.*



Slika 1.12. Sistematizacija numeričkih jednodimenzionalnih metoda za NP

Metode eliminacije mogu se koristiti i za optimizaciju diskontinuiranih funkcija. Kvadratične metode aproksimiraju datu funkciju polinomom drugog dok kubne metode polinomom trećeg stupnja.

Kako bi se mogao objasniti način rada pojedinih numeričkih jednodimenzionalnih metoda potrebno je objasniti što su to unimodalne funkcije.

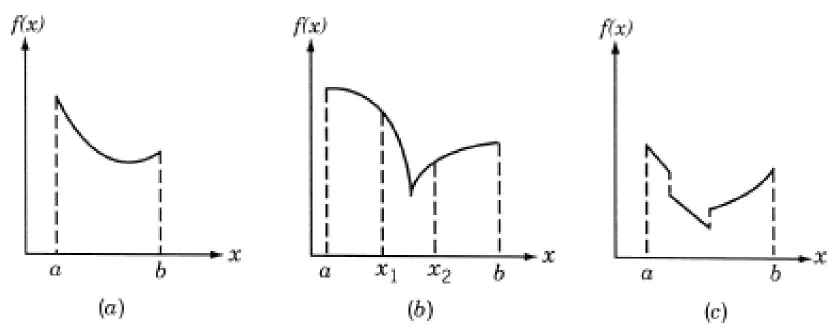
Unimodalne funkcije

Unimodalna funkcija je funkcija koja ima samo jedan maksimum (vrh) ili minimum na zadanom intervalu promatranja. Za unimodalne jednovarijabilne funkcije može se reći da ako postoje dvije točke x_1 i x_2 na istoj strani u odnosu na optimum tada ona koja je bliža optimumu daje boju vrijednost funkcije cilja od one druge.

Ova formulacija može se matematički sročiti na sljedeći način:

Funkcija $f(x)$ zadovoljava uvjet unimodalnosti ako za $x_1 < x_2 < x^*$ vrijedi $f(x_2) < f(x_1)$ kao i da vrijedi ako je $x_2 > x_1 > x^*$ vrijedi $f(x_1) < f(x_2)$, x^* je točka minimuma.

Na slici 1.13. prema [1] prikazane su tri unimodalne funkcije od kojih je funkcija (a) derivabilna, funkcija (b) nederivabilna a funkcija (c) diskontinuirana.



Slika 1.13. Unimodalne funkcije

Ako je poznato da je funkcija unimodalna na nekom području, interval u kojem se nalazi optimum može se sužavati postepeno što je osnova rada eliminacijskih metoda .

Na slici 1.14. prema [1] prikazani su rezultati dvaju pokusa koji su rezultirali procjenama iznosa funkcije cilja u točkama x_1 i x_2 za tri slučaja (a), (b) i (c) uz pretpostavku da je funkcija cilja unimodalna.

Područje mogućih rješenja varijable x iznosi od 0 do 1, pri čemu su moguća tri ishoda pokusa za sva tri slučaja a to su da je $f_1 < f_2$, $f_1 > f_2$ i da su $f_1 = f_2$.

Slučaj (a)

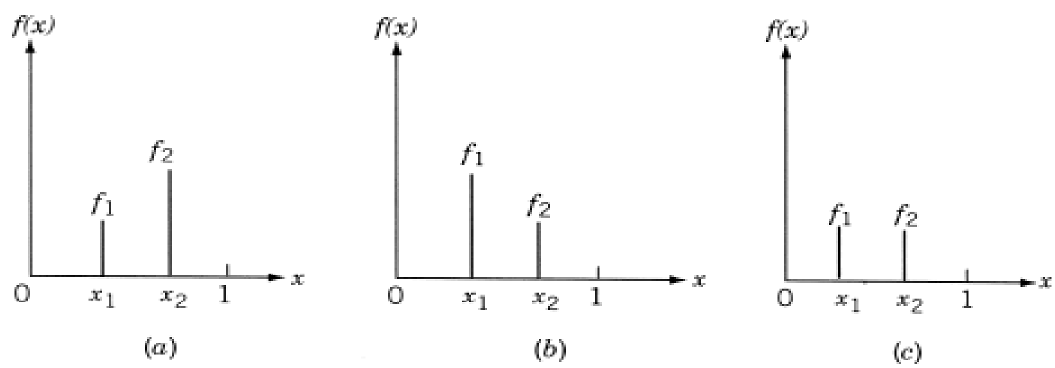
Interval $[x_2,1]$ se odbacuje

Slučaj (b)

Interval $[0,x_1]$ može se otpisati jer je $f_1 > f_2$ tako da preostaje manji suženi interval $[x_1,1]$ u kojem se nalazi optimum.

Slučaj (c)

U ovom slučaju $f_1 = f_2$ tako da se otpisuju intervali $[0,x_1]$ i $[x_2,1]$ nakon čega preostaje novi suženi interval nesigurnosti odnosno interval u kojem je optimalno rješenje.



Slika 1.14. Ishod dvaju pokusa za slučajeve (a) $f_1 < f_2$; (b) $f_1 > f_2$ i (c) $f_1 = f_2$.

METODE ELIMINACIJE

1. 4.1. Metoda neograničene pretrage sa nepromjenjivim iznosom koraka pretrage

Osnovni pristup ove metode je da se krene sa potragom iz neke početne točke sa fiksnim iznosom koraka pretrage u odgovarajućem smjeru koji može biti pozitivan i negativan u sljedeću točku i tako sve dok se ne utvrdi optimum .

Karakteristika ove metode je to što na konačnu preciznost s kojom se utvrđuje optimum utječe veličina izabranog koraka pretrage. Što je on manji to je točnost veća ali je potrebno više vremena da bi se proveli potrebni proračuni .Ova metoda lako se provodi ali u mnogo slučajeva nije dovoljno učinkovita.

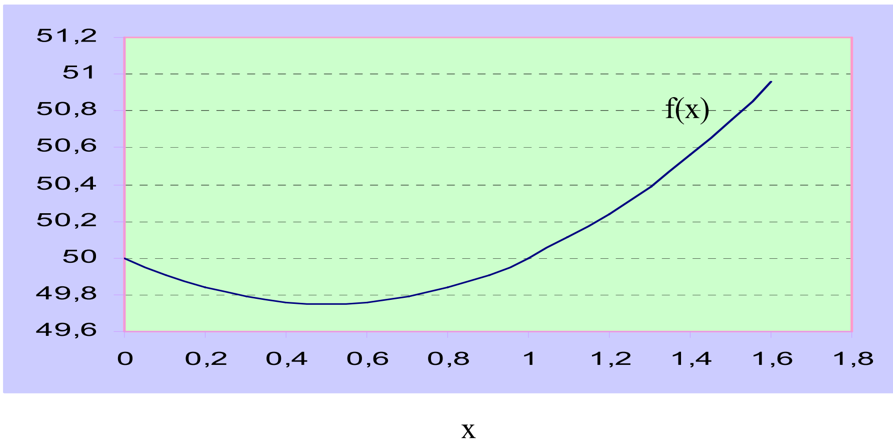
Nedostatak se može vidjeti ako je npr. potrebno odrediti minimum neke funkcije koji se nalazi u točki x_{opt} = 70 000 uz početnu točku $x_1 = 0$ i korak pretrage $s = 0.1$.Da bi metoda odredila optimum potrebno je provesti 7 000 001 iteraciju u kojoj se računa vrijednost funkcije cilja, to zahtjeva veliku količinu računanja koje računalo mora provesti.
Ovaj i slični problemi posebno se ističu kod pronalaženja optimuma u problemima bez ograničenja.

Koraci u kojima se metoda sa nepromjenjivim korakom provodi:

- Odabir početne točke pretrage x_1
- Pronaći iznos funkcije cilja u početnoj točki $f_1=f(x_1)$
- Odabir koraka pretrage s i utvrđivanje sljedeće točke kao $x_2=x_1+s$
- Pronaći iznos funkcije cilja u točki x_2 kao $f_2=f(x_2)$
- Ako se traži minimum i ako je $f_2 < f_1$ uz pretpostavku unimodalnosti poznato je tada da minimum ne može biti smješten u intervalu gdje je $x < x_1$. Pretraga se nastavlja iz točke x_2 u sljedeću točku i tako sve dok se funkcija cilja smanjuje, kada pretraga dođe u točku $x_i = x_1+(i-1)\cdot s$ u kojoj je funkcija cilja veća od funkcije cilja u prethodnoj točki
- Kada se pronađe točka x_i u njoj se zaustavlja i točka x_{i-1} se usvaja kao optimalno rješenje
- Ako je $f_2 > f_1$ tada se pretraga nastavlja u drugom smjeru i pronalaze se točke x_{-2}, x_{-3}, x_{-4} itd. Općenito vrijedi $x_{-j} = x_1-(j-1)\cdot s$
- Ako je $f_2 = f_1$ minimum je smješten između x_1 i x_2
- Ako su f_2 i f_{-2} jednake tad je minimum smješten u dvostrukom intervalu $x_2<x<x_2$

Npr. potrebno je pronaći minimum funkcije $f(x) = x^2 - x + 50$ koja je očigledno unimodalna prikazane na slici 1.15. uz veličinu koraka s=0.1 i početnu točku pretrage x_1=0.

Problem je riješen uz pomoć Microsoft Excell programa kako bi se uštedjelo na vremenu koje bi bilo potrebno za ručno računanje pri čemu je generirana tablica 1.1.



Slika 1.15. Unimodalna funkcija $f(x) = x^2 - x + 50$

Objašnjenje tablice 1.1.

Uz pretpostavku unimodalnosti prvo se vršio izračun funkcije za dvije točke ljevo i desno pomaknute u odnosu na početnu točku. Vidi se da u negativnom smjeru u točki x_{-1}=-0,1 funkcija raste a u točki x_1=0.1 funkcija pada stoga se pretraga nastavlja u i smjeru odnosno u desno. Iteracije su se vršile skroz dok funkcija cilja nije počela rasti u točki x_6 stoga se točka x_5 usvaja kao optimalna točka u kojoj funkcija cilja iznosi $f_5=f(x_5)$=49,75

Tablica 1.1. Proračun metodom neograničene pretrage sa stalnim korakom

x_{-i}	x_i	$f\,(x_i)$	$f\,_{(x\text{-}i)}$
	0	50	50
-0,1	0,1	49,91	50,11
-0,2	0,2	49,84	50,24
-0,3	0,3	49,79	50,39
-0,4	0,4	49,76	50,56
-0,5	0,5	49,75	50,75
-0,6	0,6	49,76	50,96
-0,7	0,7	49,79	51,19

Uz korak s=0.01 optimalno rješenje koje se dobije nakon 50 iteracija iznosi :

f_{50} = 49.7500 tako da je usporedbi sa manjim korakom s=0.1 postignuta 0.02% veća točnost.

1. 4.2. Metoda neograničene pretrage sa promjenjivim iznosom koraka pretrage

Ove metode koriste dvostruko povećavanje koraka sve dok funkcija cilja u narednim točkama raste . Kada se dostigne na taj način točka x_{i-1} koja ima najmanju vrijednost funkcije cilja tada se duplo smanjuje korak i započinje se nova pretraga iz početne točke x_i ili iz x_{i-1}. Ako se ne postigne bolji rezultat korak se opet smanjuje duplo i ta procedura se ponavlja sve dok se ne postigne bolji rezultat. Ako se ne postigne bolje rješenje korak se smanjuje sve do unaprijed određene veličine nakon čega ako se ne pronđe bolje rješenje usvaja se točka x_{i-1} kao optimum. U sljedećoj tablici prikazan je primjer proračuna za

funkciju $f(x) = x^2 - x + 50$.

Tablica 1.2. Proračun metodom neograničene pretrage sa stalnim korakom

Iteracija	s	$x_i = x_1+s$	$f_i=(x_i)$	$f_i > f_i$-1?	$x_{-i} = x_1-s$	$f_{-i}=(x_{-i})$	$f(x_{-i}) > f(x_{-i})$-1 ?
1	0,005	0	50		0		
2	0,01	0,005	49,99503	Ne	-0,005	50,00503	Da
3	0,02	0,015	49,98523	Ne	-0,015		STOP
4	0,04	0,035	49,96623	Ne	-0,035		
5	0,08	0,075	49,93063	Ne	-0,075		
6	0,16	0,155	49,86903	Ne	-0,155		
7	0,32	0,315	49,78423	Ne	-0,315		
8	0,64	0,635	49,76823	Ne	-0,635		
9	1,28	1,275	50,35063	Da	-1,275		

0.64/2=0.32

Iteracija	s	$x_i = x_1+s$	$f_i=(x_i)$	$f_i > f_i$-1?	$x_{-i} = x_1-s$	$f_{-i}=(x_{-i})$	$f(x_{-i}) > f(x_{-i})$-1 ?
1	0,32	0,635	49,76823		0,635	49,76823	
2	0,16	0,955	49,95703	Da	0,315	49,78423	Da
				STOP			STOP

Iteracija	s	$x_i = x_1+s$	$f_i=(x_i)$	$f_i > f_i$-1?	$x_{-i} = x_1-s$	$f_{-i}=(x_{-i})$	$f(x_{-i}) > f(x_{-i})$-1 ?
1	0,16	0,635	49,76823		0,635	49,76823	
2	0,08	0,795	49,83703	Da	0,475	49,75063	Ne
				STOP			

0.08 /2=0.04

Iteracija	s	$x_i = x_1+s$	$f_i=(x_i)$	$f_i > f_i$-1?	$x_{-i} = x_1-s$	$f_{-i}=(x_{-i})$	$f(x_{-i}) > f(x_{-i})$-1 ?
1	0,04	0,475	49,75063		0,475	49,75063	
2	0,02	0,515	49,75023	Da	0,435	49,75423	Da
				STOP			STOP

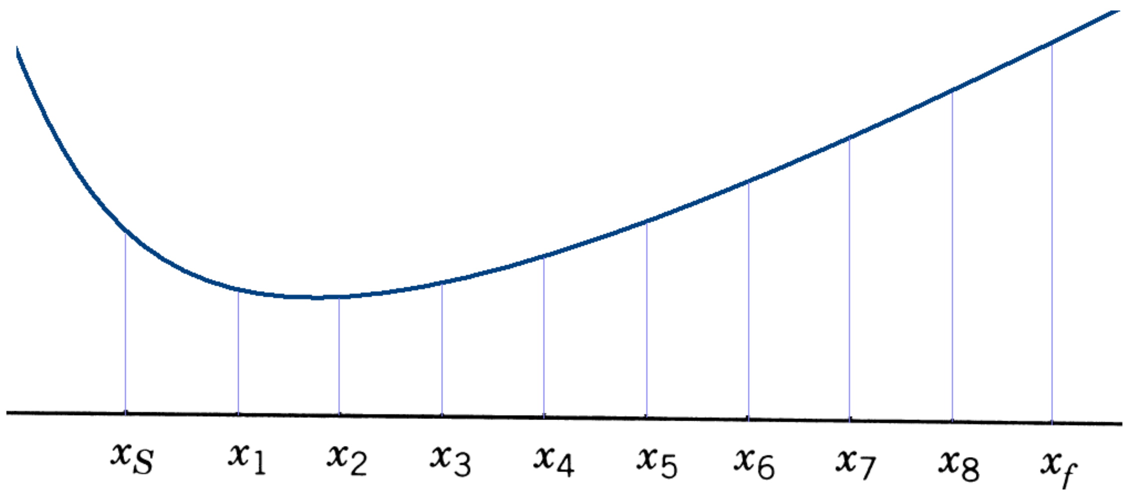
0.02 /2=0.01

Iteracija	s	$x_i = x_1+s$	$f_i=(x_i)$	$f_i > f_i$-1?	$x_{-i} = x_1-s$	$f_{-i}=(x_{-i})$	$f(x_{-i}) > f(x_{-i})$-1 ?
1	0,01	0,475	49,75063		0,475	49,75063	
2	0,005	0,485	49,75023	Ne	0,465	49,75123	Da
		OPTIMUM					
				STOP			STOP

Optimalno rješenje je točka x=0,485 u kojoj funkcija cilja iznosi 49,75023

1. 4.3. Metoda iscrpne pretrage

Metoda iscrpne pretrage ne koristi se u slučaju kada nije poznat interval u kojem se nalazi rješenje. Metoda radi na principu da izračunava vrijednost funkcije cilja za svaku od točaka koje su jednako udaljene međusobno. Te točke se određuju unaprijed i leže na intervalu od početne točke x_s i krajnje točke x_f. Uz pretpostavku unimodalnosti postepeno se smanjuje interval nesigurnosti.



Slika 1.16. Načelo rada metode iscrpne pretrage

Na slici 1.16. prema [1] prikazana je funkcija cilja i osam jednako udaljenih točaka u kojima se računa njen iznos. Iz slike se može vidjeti da je optimalno rješenje negdje između x_1 i x_3 tako da to područje predstavlja novi interval nesigurnosti koji se opet dijeli na n jednakih dijelova, taj postupak se ponavlja n puta tako da imamo *n* proračuna i *n* intervala nesigurnosti.

Općenito iznos funkcije cilja se računa u n jednako udaljenih točaka pri čemu je početni interval nesigurnosti $L_0=x_f - x_s$, ako je optimalno rješenje točka x_j tada je konačni interval nesigurnosti nakon *n* iznosi:

$$L_n = x_{j+1} - x_{j-1} = \frac{2}{n+1} L_0$$

Tablica 1.3. Iznosa intervala nesigurnosti za svaki od ukupno *n* postupaka

Postupak br.	2	3	4	5	....	*n*
L_n / L_0	2/3	2/4	2/5	2/6	....	2/ (*n*+1)

Ako se uzme središnja točka *n*-tog intervala neigurnosti kao optimalno rješenje maksimalna devijacija može biti *Lo/(n+*1*)* stoga kako bismo pronašli optimum na području koje je p-ti dio ukupnog intervala nesigurnosti vrijedi izraz:

$$\frac{1}{n+1} \le \frac{1}{p} \rightarrow n \ge p-1$$ gdje se p obavezno uvrštava u postocima

U tablici 1.4. prikazan je primjer pronalaženja optimalnog rješenja funkcije

$f(x) = x^2 - x + 50$ uz početnu točku $x_s = -10$, krajnju točku $x_f = 10$ i p = 11% (0,11)

$n \ge 11-1 \rightarrow n=10$, $L_0=10-(-10)=20$

Tablica 1.4.Primjer proračuna metodom iscrpne pretrage

n=1 razmak između točaka = $L_0 / (n+1)$=20/10 = 1,8181

	xs	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	xf
x	-10	-8,2	-6,36	-4,5455	-2,72727	-0,9091	0,90909	2,727273	4,5455	6,364	8,18	10
f(x)	160	125	96,9	75,207	60,16529	51,7355	49,9174	54,71074	66,116	84,13	109	140

n=2 razmak između točaka = $L_0 / (n+1)$=20/10 = 0,3306

	xs	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	xf
x	-0,91	-0,6	-0,25	0,0826	0,413223	0,7438	1,07438	1,404959	1,7355	2,066	2,4	2,73
f(x)	51,74	50,9	50,3	49,924	49,75753	49,8094	50,0799	50,56895	51,277	52,2	53,3	54,7

n=3 razmak između točaka = $L_0 / (n+1)$=20/10 = 0,00601

	xs	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	xf
x	0,083	0,14	0,2	0,263	0,323065	0,38317	0,44328	0,503381	0,5635	0,624	0,68	0,74
f(x)	49,76	49,9	49,8	49,806	49,78131	49,7636	49,7532	49,75001	49,754	49,77	49,8	49,8

n=4 razmak između točaka = $L_0 / (n+1)$=20/10 = 0,00109

	xs	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	xf
x	0,443	0,45	0,47	0,4761	0,486989	0,49792	0,50884	0,519773	0,5307	0,542	0,55	0,56
f(x)	49,75	49,8	49,8	49,751	49,75017	49,75	49,7501	49,75039	49,751	49,75	49,8	49,8

Svjetlo plavom bojom označena su optimalna rješenja za svaki korak dok su svjetlo zelenom bojom označene granice intervala za sljedeći korak.
Nakon *n*=3 metoda je došla do rješenja stoga nema potrebe provoditi daljnje proračune jer je već postignuta zadovoljavajuća točnost.
U tablici je proveden proračun za *n*=4 kako bi se dokazalo da se optimum ne mijenja znatno.

1.5. Optimizacija NP problema sa ograničenjima

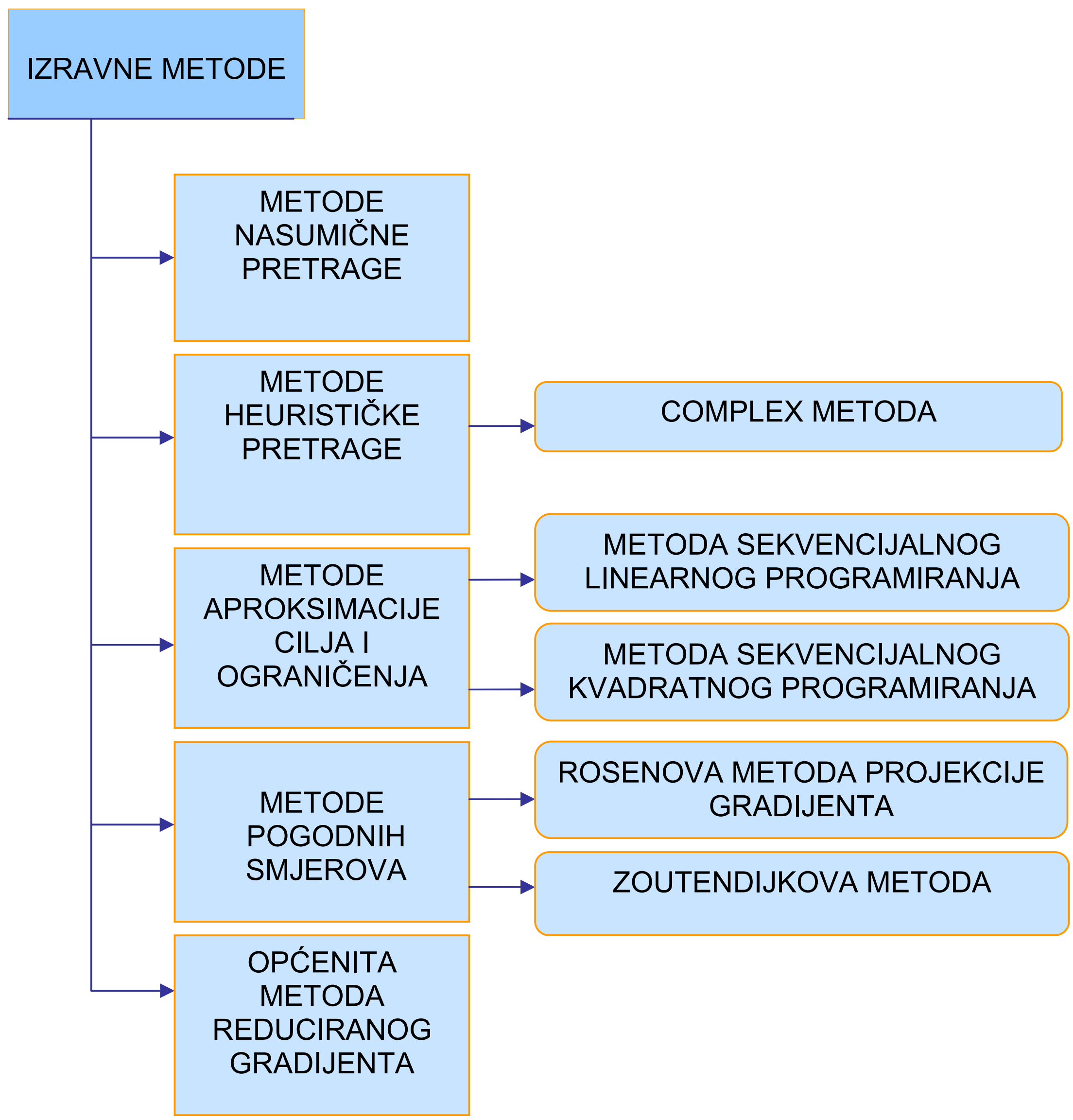
Metode za rješavanje NP problema sa ograničenjima dijele se na izravne i neizravne metode.

1.5.1. IZRAVNE METODE

Podjela izravnih metoda prikazana je na slici 1.15. prema kojoj se one dijele na metode nasumične pretrage, metode heurističke pretrage, metode aproksimacije cilja i ograničenja, metode pogodnih smjerova i općenitu metodu reduciranog gradijenta.

Metoda aproksimacije cilja i ograničenja dijeli se na metodu sekvencijalnog linearnog programiranja (SLP metodu) i metodu sekvencijalnog kvadratnog programiranja.

Rosenova metoda projekcije gradijenta i Zoutendijkova metoda zajedno spadaju u metode reduciranog gradijenta.



Slika 1.17. Izravne metode za probleme NP sa ograničenjim

1. METODA NASUMIČNE PRETRAGE

Princip rada metode je:

- Generiranje probnog vektora na način da se koristi generiranje slučajanih brojeva za svaku varijablu vektora.
- Provjerava se je su li zadovoljena ograničenja za probni vektor. Ako se radi o ograničenjima sa jednakostima provjerava se jesu li veličine u ograničenjima u nekim zadanim tolerancijskim poljima. Ako su narušena ograničenja tada se generira novi probni vektor.
- Ako novi vektor zadovoljava ograničenja provjerava se uzrokuje li on manju vrijednost funkcije cilja od prijašnjeg probnog vektora koji je zadovoljio ograničenja, ako je to slučaj tada se novi probni vektor usvaja kao najbolje dosadašnje rješenje.
- Postupak se ponavlja unaprijed određeni broj puta nakon čega se pretraga prekida i usvaja se dotad najbolji pogodni vektor kao optimalno rješenje.

Uz ovaj osnovni postupak moguće je napraviti nekoliko modifikacija koje znatno povećavaju učinkovitost i točnost metode. Primjer je metoda nasumične pretrage koja bi nakon pronalaska pogodnog probnog vektora koristila generiranje slučajnih brojeva kako bi se odredio pogodan smjer kretanja za određivanje novog probnog vektora. Kada se otkrije pogodan smjer kretanja tada se jednodimenzionalnom pretragom pronalazi sljedeći pogodan probni vektor.

Općenito metode nasumične pretrage nisu učinkovite u usporedbi s drugim metodama ali imaju određenih prednosti pred njima kao što su:

- Lako se programiraju
- Pouzdane su u pronalaženju rješenja bliskog optimumu ako se provede dovoljno velik broj iteracija
- Pronalaze globalni optimum čak i kada se radi o nekonveksnim problemima što je najveća prednost pred nekim od drugih metoda

Primjer 1:

Potrebno je pronaći metodom nasumične pretrage vrijednost varijabli koje uzrokuju minimalnu vrijednost funkcije cilja $F(X)=x_1 \cdot x_2 - 8 \cdot x_1 + 5 \cdot x_2$ podložno ograničenjima:

$$x_1 < x_2$$

$$0 \le x_1 < 10$$

$$0 \le x_2 < 10$$

Pomoću programa Microsoft Excell generirano je 7000 slučajnih brojeva za određivanje 7000 vrijednosti svake varijable pri čemu se vršio proračun funkcije cilja bez i sa ograničenjima. U predzadnjem stupcu tablice 1.5. nalaze se vrijednosti funkcije cilja za iznose varijabli koje nemoraju zadovoljavati rješenja dok su u zadnjem stupcu filtrirane vrijednosti funkcije cilja iz predzadnjeg stupca koje zadovoljavaju sva ograničenja.

Tablica 1.5. Primjer proračuna metodom nasumične pretrage

iteracija	$sl.br._1$	x_1	$sl.br._2$	x_2	$F(x_1,x_2)$	$F(x_1,x_2)$ OK
1	0,434723	4,3	0,306881	3	-6,5	0
2	0,379985	3,7	0,384421	3,8	3,46	3,46
3	0,695327	6,9	0,460116	4,6	-0,46	0
4	0,989399	9,8	0,068915	0,6	-69,52	0
5	0,079563	0,7	0,730154	7,3	36,01	36,01
6	0,904609	9	0,8319	8,3	44,2	0
7	0,362235	3,6	0,630296	6,3	25,38	25,38
8	0,687542	6,8	0,046586	0,4	-49,68	0
9	0,860258	8,6	0,402034	4	-14,4	0
10	0,04995	0,4	0,984863	9,8	49,72	49,72
11	0,918919	9,1	0,961415	9,6	62,56	62,56
12	0,024448	0,2	0,361833	3,6	17,12	17,12
13	0,664458	6,6	0,163044	1,6	-34,24	0
14	0,522196	5,2	0,191881	1,9	-22,22	0
15	0,283051	2,8	0,70786	7	32,2	32,2
16	0,748981	7,4	0,951682	9,5	58,6	58,6
17	0,960252	9,6	0,999258	9,9	67,74	67,74
18	0,966326	9,6	0,039985	0,3	-72,42	0
19	0,017147	0,1	0,418959	4,1	20,11	20,11
20	0,593883	5,9	0,701768	7	29,1	29,1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
7000	0,460003	4,6	0,285449	2,8	-9,92	0

Rješenje			
iteracija	x_1	x_2	fc_{opt}
3719	1,7	1,8	-1,54

Optimalno rješenje je *Fc*=-1,54 uz iznose varijabli x_1=1,7 i x_2=1,8 .

Program je napisan na taj način da se generiraju slučajni brojevi sa samo jednim decimalnim mjestom. Moguće je generirati slučajne brojeve sa većim brojem decimalnih mjesta ali što je veći broj decimala slučajnih brojeva to je više iteracija potrebno izvršiti kako bi se pronašao optimalan par varijabli. U ovom primjeru budući da su ograničenja postavljena tako da varijable mogu biti u zadanom intervalu uz pretpostavku o jednoj decimali svaka varijabla je mogla poprimiti 100 mogućih iznosa i to od 0 do 9,9.

Na broj potrebnih iteracija koje je potrebno provesti utječe broj varijabli i broj decimalnih mjesta varijabli odnosno zahtjevana preciznost rješenja. Kako bi se dijelom ublažio ovaj problem moguće je napisati takav program koji bi ugrubo našao rješenje i zatim u njegovoj okolini započeo novu pretragu kako bi se postigla zadovoljavajuća točnost.

U sljedećem primjeru riješen je nelinearni problem kojem već poznajemo optimalno rješenje u svrhu usporedbe rješenja dobivenog metodom iscrpne pretrage sa optimalnim rješenjem.

Primjer 2:

Potrebno je pronaći maksimum funkcije $F(X)=2x_1-x_1^2+x_2$ uz ograničenja :

(1) $2x_1^2+3x_2^2 \leq 6$

(2) $x_1 \geq 0$

(3) $x_2 \geq 0$

Postupak rješavanja u Microsoft Excellu:

- Potrebno je načiniti tablicu u kojoj je definirano pravilo prema kojem se određeni raspon slučajnih brojeva pomoću funkcije „LOOKUP“ koristi za generiranje iznosa varijabli. U tablici 1.6. vrijednost upisana u žuto polje određuje koliki će biti raspon vrijednosti varijabli (kreće se od 0 do 10) . Ako želimo postići veću preciznost vrijednosti u stupcu „Vrijednost slučajnog broja“ trebale bi se mjenjati za manje iznose npr. 0,015, 0,02, 0,025,...10 odnosno treba smanjiti korak. U tom slučaju potrebno bi bilo proširiti tablicu da ima veći broj redaka.

Tablica 1.6. Raspon vrijednosti varijabli i preciznost

lookup	Vrijednost slučajnog broja (korak k =0,01)	Vrijednost varijable
10	0	0
	0,01	0,1
	0,02	0,2
	0,03	0,3
	0,04	0,4
	0,05	0,5
	0,06	0,6
	0,07	0,7
	⋮	⋮
	1	10

$$\boxed{10 \cdot 0{,}01 = 0{,}1}$$

- Kada se načini tablica raspona potrebno je generirati slučajne brojeve za obje varijable.

  Vrijednosti slučajnih brojeva uspoređuje se sa vrijednostima u tablici raspona.

  Kada je slučajni broj

  U ovom primjeru generirano je 5000 slučajnih brojeva.

Tablica 1.7. Generiranje slučajnih brojeva i iznosa varijabli

Iteracija	*Sl.br.* $_1$	*Sl.br.* $_2$	x_1	x_2
1	0,0532	0,9739	0,5	9,7
2	0,1959	0,7466	1,9	7,4
3	0,9122	0,7622	9,1	7,6
4	0,2841	0,5289	2,8	5,2
5	0,2882	0,3121	2,8	3,1
6	0,7371	0,6636	7,3	6,6
7	0,6622	0,126	6,6	1,2
8	0,8619	0,8402	8,6	8,4
⋮	⋮	⋮	⋮	⋮
5000	0,8619	0,8402	8,6	8,4

- Računaju se vrijednosti funkcije cilja za svaki redak odnosno za svih 5000 parova varijabli, nakon čega je potrebno oformiti jedan stupac „ogr OK“ koji služi za detekciju jesu li ispunjena ograničenja ili ne. Ako su ograničenja ispunjena vrijednosti u stupcu poprimaju iznos 1 a ako nisu tada poprimaju vrijednost 0.

Tablica 1.8. Generiranje slučajnih brojeva i iznosa varijabli

i	*Sl.br.* $_1$	*Sl.br.* $_2$	x_1	x_2	F_c	*ogr*	*ogr ok*	*fc 0k*
1	0,0532	0,9739	0,5	9,7	10,45	282,77	0	0
2	0,1959	0,7466	1,9	7,4	7,59	171,5	0	0
3	0,9122	0,7622	9,1	7,6	-57,01	338,9	0	0
4	0,2841	0,5289	2,8	5,2	2,96	96,8	0	0
5	0,2882	0,3121	2,8	3,1	0,86	44,51	0	0
6	0,7371	0,6636	7,3	6,6	-32,09	237,26	0	0
7	0,6622	0,126	6,6	1,2	-29,16	91,44	0	0
8	0,8619	0,8402	8,6	8,4	-48,36	359,6	0	0
...	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
5000	0,8619	0,8402	8,6	8,4	-48,36	359,6	0	0

- Filtriraju se one vrijednosti funkcije cilja kod koji iznosi varijabli zadovoljavaju ograničenja . To se postiže na način da se formira stupac nazvan podobnih rješenja koja zadovoljavaju ograničenja „fc OK“ čije vrijednosti se računaju tako da se vrijednosti funkcije cilja pomnože sa vrijednostima u stupcu provjeravanja ograničenja.
- Formira se tablica rješenja koja filtrira iz stupca podobnih ograničenja maksimalno rješenje funkcijom MAX( ) te se pomoću funkcije VLOOKUP() određuje u kojoj je to iteraciji postignuto i koji su iznosi varijabli u njoj.

Tablica 1.9. Tablica prvog riješenja

Rješenje 1	Fc_{opt}	iteracija	x_1	x_2
	2,16	1258	0,8	1,2

- Smanjuje se raspon pretrage sa 10 na 1,5 pri čemu se postižu sljedeći rezultati:

Rješenje 2	Fc_{opt}	iteracija	x_1	x_2
	2,20478	2514	0,735	1,275

- Smanjuje se raspon pretrage sa 1.5 na 1.3 pri čemu se postižu sljedeći rezultati:

Rješenje 3	Fc_{opt}	iteracija	x_1	x_2
	2,2126	3007	0,78	1,261

- Smanjivanjem raspona pretrage više nije moguće postići znatno poboljšanje rješenja.Ukoliko se želi postići još bolje rješenje potrebno je povećati preciznost i što zahtjeva ponovno programiranje tablice raspona i preciznosti i modifikaciju formula za generiranje vrijednosti varijabli na temelju slučajnih brojeva.

Nakon smanjivanja koraka sa k= 0,01 na k=0,005 nakon 5000 iteracija uz zadržavanje raspona R_p=1,3 postignuti su sljedeći rezultati:

Rješenje 4	Fc_{opt}	iteracija	x_1	x_2
	1,86249	1	1,2025	0,9035

Komentar

Za očekivati je bilo da će se postići poboljšanje rješenja s obzirom na manji raspon varijabli i manji korak što nije postignuto tako da se iz toga vidi da broj iteracija nije bio dovoljan da bi se generiralo bolje rješenje.

Ako bi se zadržao manji korak i veći raspon postignuti su bolji rezultati :

Rješenje 4	Fc_{opt}	iteracija	x_1	x_2
	2,18438	3000	0,875	1,2

Optimalno rješenje je pronađeno pomoću metode ograničenog gradijenta i iznosi:

F_{max}= 2,2136 i X = (x_1;x_2) = (0,8108;1,2494) dok je metodom nasumične pretrage uz raspon pretrage R= 1.3 i korak k=0.01 pronađeno rješenje F (X)= 2,2126 i X = (x_1;x_2) = (0,780;1,261)

Ovaj program dokazano ovisi o broju iteracija jer iz gornjih primjera vidi se da se smanjivanjem raspona i koraka ne postižu nužno bolji rezultati.

Da bi se problem bolje riješio moguće je pristupiti generiranju vrijednosti varijabli na drugačiji način tako da se formira tablica raspona za svaku varijablu u kojima bi se vrijednosti varijabli kretale oko zadnjih najboljih vrijednosti umjesto da se generiraju od nule do 1.3. To se može shvatiti kao početak nove pretrage u točki (0,780;1,261).

Kako se u Excellu slučajni brojevi kreću od o do 1 tada se slučajni broj 0,5 pridjeljuje kao iznos varijable x_1=0,78 i x_2= 1,261

Tablica 1.10. Različiti rasponi za varijable

0,001	Vrijednost slučajnog broja	Vrijednost varijable x_1	Vrijednost varijable x_2
100	0	0,78-100·0,001	1,261-100·0,001
99	0,05	0,78-99·0,001	1,261-99·0,001
	⋮	⋮	⋮
2	0,49	0,78-2·0,002	1,261-2·0,001
1	0,495	0,78-1·0,001	1,261-1·0,001
0	0,5	0,78	1,261
1	0,505	0,78+1·0,001	1,26+1·0,001
2	0,51	0,78+2·0,001	1,261+2·0,001
	⋮	⋮	⋮
100	1	0,78+100·0,001	1,261+100·0,01

Rješenja dobiveno pretragom oko točke (0,780;1,261) glasi:

rj	*F*c	iteracija	x_1	x_2
	2,21372	1682	0,778	1,263

Optimalno rješenje pronađeno pomoću metode ograničenog gradijenta iznosi:

F_{max}= 2,2136 i X = (x_1;x_2) = (0,8108;1,2494) dok je dobiveno rješenje čak i bolje što dokazuje da je ova metoda uz određene modifikacije vrlo učinkovita .

2. COMPLEX METODA

Ova metoda poznata još kao Complex Box metoda je zapravo modificirana simplex metoda koja može rješavati NP probleme sa ograničenjima koja su nelinearna.

Općenito potrebno je minimizirati funkciju F(X) podložnu ograničenjima oblika:

$g_j(X) \leq 0, \quad j=1,2,3...m$

$x_i^{(d)} \leq x_i \leq x_i^{(g)}, \quad i=1,2,3...n$ (bočna ograničenja)

Često je slučaj da se zadovoljenje ograničenja raspona varijable x_i ne podudara sa ograničenjima $g_j(X) \leq 0$.

Kao i kod klasične simplex metode i complex metoda na sličan način sljedno formira poliedre koji imaju $k_v \geq n_v+1$ broj vrhova, gdje je n_v broj varijabli.

Princip rada metode je:

- Pronaći $k_v \geq n_v+1$ vrhova tj. točaka koje zadovoljavaju svih m ograničenja. To se provodi u praksi tako da se pronađe početna pogodna točka X_1 (ona koja zadovoljava ograničenja) a tada preostalih k_v-1 točaka koje se generiraju jedna po jedna koristeći generiranje slučajnih brojeva u rasponu od 0 do 1 prema izrazu:

$$x_{i,j}=x_i^{(d)}+r_{i,j}(x_i^{(g)}-x_i^{(d)}),\ i = 1,2....n_v,\ j = 2,3...k$$

gdje je $x_{i,j}$ i-ta komponenta točke X_j a $r_{i,j}$ slučajni broj na intervalu [0,1] . Točke $X_2, X_3, ...X_k$ zadovoljavaju bočna ograničenja ali nemoraju zadovoljavati $g_j(X) \leq 0$ ograničenja.

Kada se pronađe nova točka X_j tada se provjerava zadovoljava li ona ograničenja, ako to nije slučaj tada se probna točka X_j pomiče na pola svoje udaljanosti prema centroidu preostalih već prihvaćenih točaka u koje spada i početna točka X_1. Jednadžba za određivanje koordinata centroida glasi:

$$X_0 = \frac{1}{j-1}\sum_{l=1}^{j-1} X_l$$

Ako probna točka i dalje krši neka od ograničenja $g_j(X) \leq 0$ nastavlja se postupak pomicanja točke na pola puta prema centroidu i ponavlja se sve dok se ne pronađe točka X_j koja zadovoljava sva ograničenja.

- Vrijednost funkcije cilja se provjerava u svakoj od k točaka. Ako je u točki X_h vrijednost funkcije cilja najveća tada se zrcaljenjem pronalazi sljedeća točka X_r prema izrazu $X_r=(1+\alpha)\cdot X_0-\alpha\cdot X_h$. Za početak uzima se vrijednost $\alpha \geq 1$, dok je X_0 centroid svih vrhova osim vrha X_h i računa se kao:

$$X_0 = \frac{1}{k-1}\sum_{\substack{l=1 \\ l\neq k}}^{k} X_l$$

- Provodi se ispitivanje zadovoljava li točka X_r ograničenja. Ako je točka pogodna i ako vrijedi da je $f(X_r)<f(X_h)$ tada se točka X_h zamjenjuje sa točkom X_r. Ako je slučaj da je $f(X_r)\geq f(X_h)$ tada se pronalazi nova probna točka X_r smanjivanjem vrijednosti α dvostruko nakon čega se opet provjerava je li $f(X_r)<f(X_h)$, ako nije tada se opet dvostruko smanjuje α i tako se postupak može ponavljati sve dok se α ne smanji do određene vrijednosti ε1. Ako poboljšana točka Xr ne zadovoljava $f(X_r)<f(X_h)$ uvijet čak ni za najmanji iznos α tada se točka Xr odbacuje i cijeli postupak zrcaljenja se ponovno ponavlja koristeći točku X_p u kojoj funkcija cilja ima drugu najveću vrijednost odmah nakon vrijednosti koju postiže u točki X_h.
- Ako u bilo kojem stadiju zrcaljena točka Xr krši bilo koje od ograničenja pomiće se dvostruko bliže centroidu dok ne postane pogodna pri čemu je:

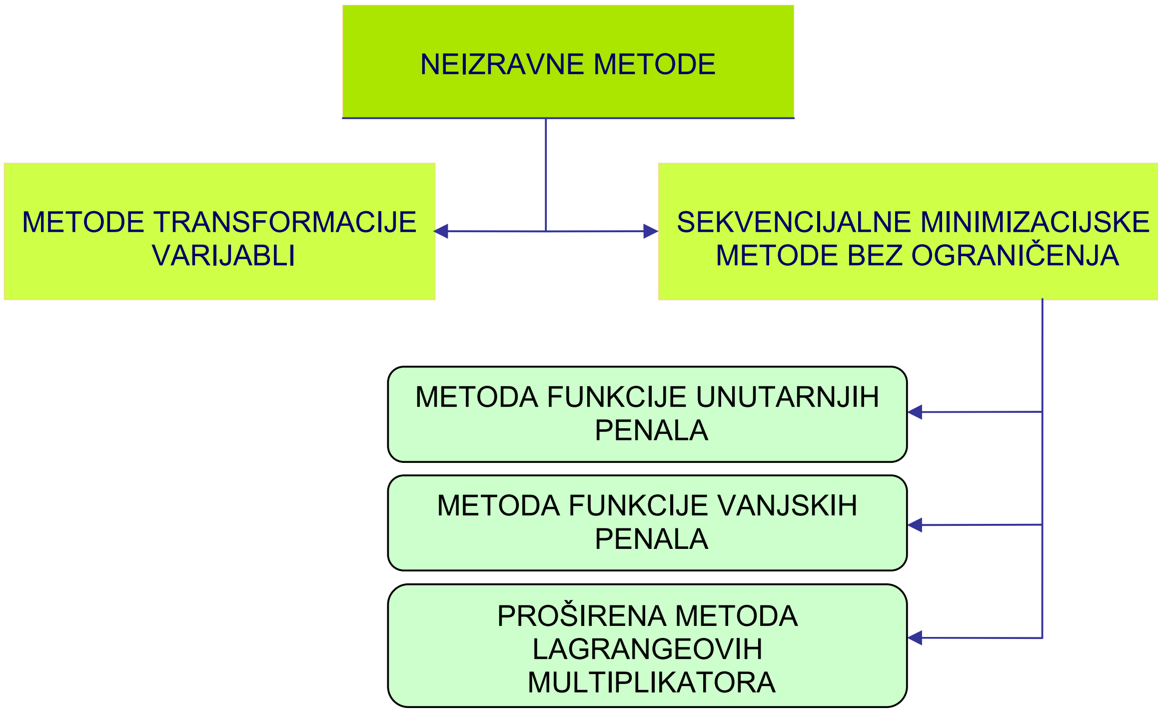
$$(X_r)_{nova} = \frac{1}{2}(X_0 + X_r)$$

Ova metoda napreduje prema optimalnoj točki sve dok se complex ne uruši u svoj centroid .

- Svaki put kada je najnepovoljnija točka Xh postojećeg complex-a zamjenjena novom točkom complex se modificira i potrebno je testirati konvergiranje procesa. Kažemo da proces konvergira prema rješenju ako su zadovoljena sljedeća dva uvjeta:
  a) Complex se smanjuje na neku malu unaprijed utvrđenu vrijednost odnosno razmak između bilo koja dva vrha od X_1,X_2...X_k se smanjuje na veličinu manju od unaprijed određene veličine ε_2.
  b) Standardna devijacija vrijednosti funkcije postaje dovoljno mala da je manja od neke unaprijed određene vrijednosti ε_3.

$$\sqrt{\frac{1}{k}\sum_{j=1}^{k}\left[f(X)-f(Xj)\right]^2} \leq \varepsilon_2$$

1.5.2. NEIZRAVNE METODE



Slika 1.18. Neizravne metode za probleme NP sa ograničenjima

Iz gornje slike vidljiva je podjela neizravnih metoda nelinearnog programiranja na metode transformacije varijabli i na sekvencijalne (slijedne) minimizacijske metode bez ograničenja koje se dalje dijele na matodu funkcije unutarnjih i metodu funkcije vanjskuh penala te na proširenu metodu Lagrangeovih multiplikatora.

Metoda transformacije varijabli zanimljiva je po tome što transformira varijable na taj način da poprimaju iznose od 0 do 1 ili od -1 do 1.

2. Sistematizacija metoda nelinearnog programiranja

2.1. Opća podjela metoda nelinearnog optimiranja

Sve metode optimiranja općenito možemo podijeliti na:

- Analitičke metode
- Numeričke metode
- Grafičke metode
- Eksperimentalne metode

Analitičke metode

Analitičke metode primjenjuju se za rješavanje problema kod kojih je funkcija cilja opisana matematičkim članovima tako da je moguće primjeniti poznata matematička pravila pri postupanju sa varijablama i funkcijama ograničenja. Nisu pogodne za rješavanje složenijih problema u nelinearnom programiranju.

Filozofija analitičkih metoda je pronalaženje one vrijednosti varijabli odnosno vektora X za koju je prva derivacija odnosno gradijent funkcije cilja f(X) jednaka nuli.
Ta točka naziva se kritična točka, nakon toga provjerava se predznak druge derivacije funkcije cilja u toj točki, ovisno o predznaku zaključuje se radi li se o minimumu ili maksimumu.

Funkcije cilja mogu imati više točaka u kojima je gradijent funkcije cilja jednak nuli tako da se optimumi dijele na globalne i lokalne optimume. Da bi se odredilo koji je od optimuma globalni optimum odnosno stvarno najbolje riješenje potrebno je izračunati vrijednosti funkcije cilja u svim kritičnim točkama i usporediti ih nakon čega se određuje stvarno rješenje.

Numeričke metode

Osnovna filozofija većine numeričkih metoda za optimizaciju nelinearnih funkcija je provođenje niza neprestanih poboljšanja aproksimacija optimalnog riješenja sljedeći naredne korake koji su načelno isti za probleme sa i bez ograničenja:

Grafičke metode

Grafičke metode pogodne su za rješavanje problema sa jednom varijablom.

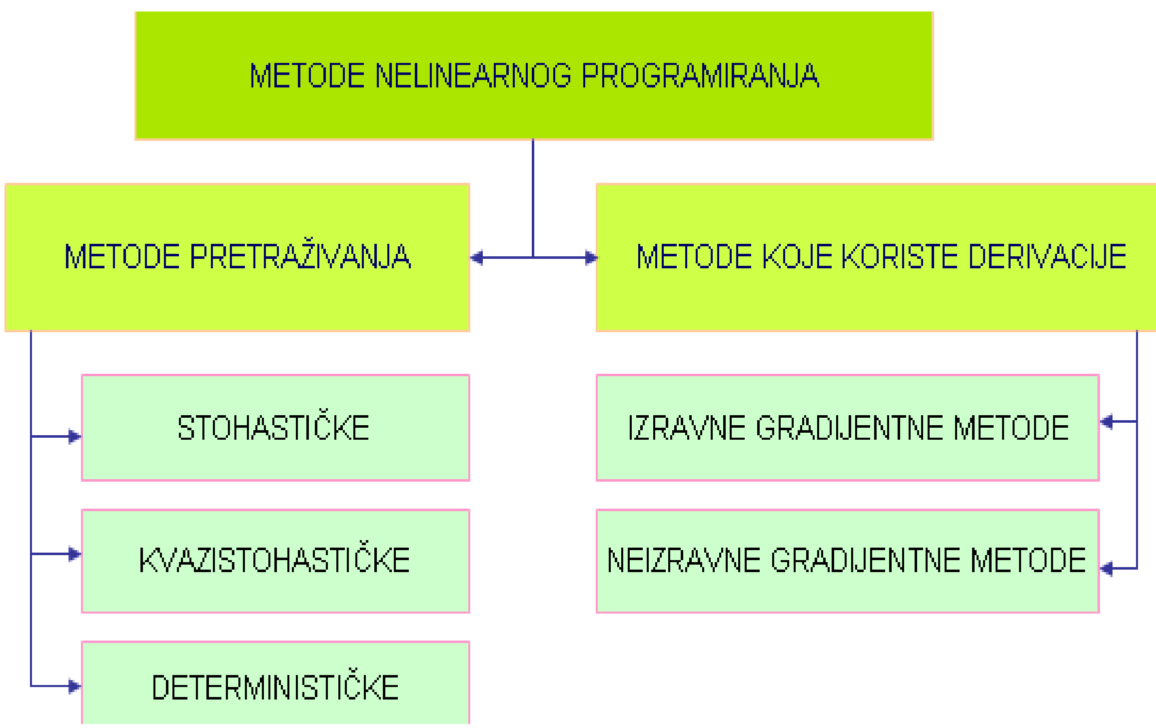
Moguće je rješavati probleme i sa dvije varijable ali to nije praktično zbog toga što je potrebno crtati u trodimenzionalnom prostoru pa se ne koristi često.

Eksperimentalne metode

Provođenjem eksperimenata nastoji se doći do ekstrema funkcija odnosno do optimuma.

Kada se provede jedan eksperiment dobiveni rezultati ključni su za određivanje gdje treba locirati sljedeći eksperiment koji će imati bolje rezultate funkcije cilja od prethodnog eksperimenta..

2.2. Podjela prema načinu rada



Slika 2.1. Podjela NP metoda prema načinu rada

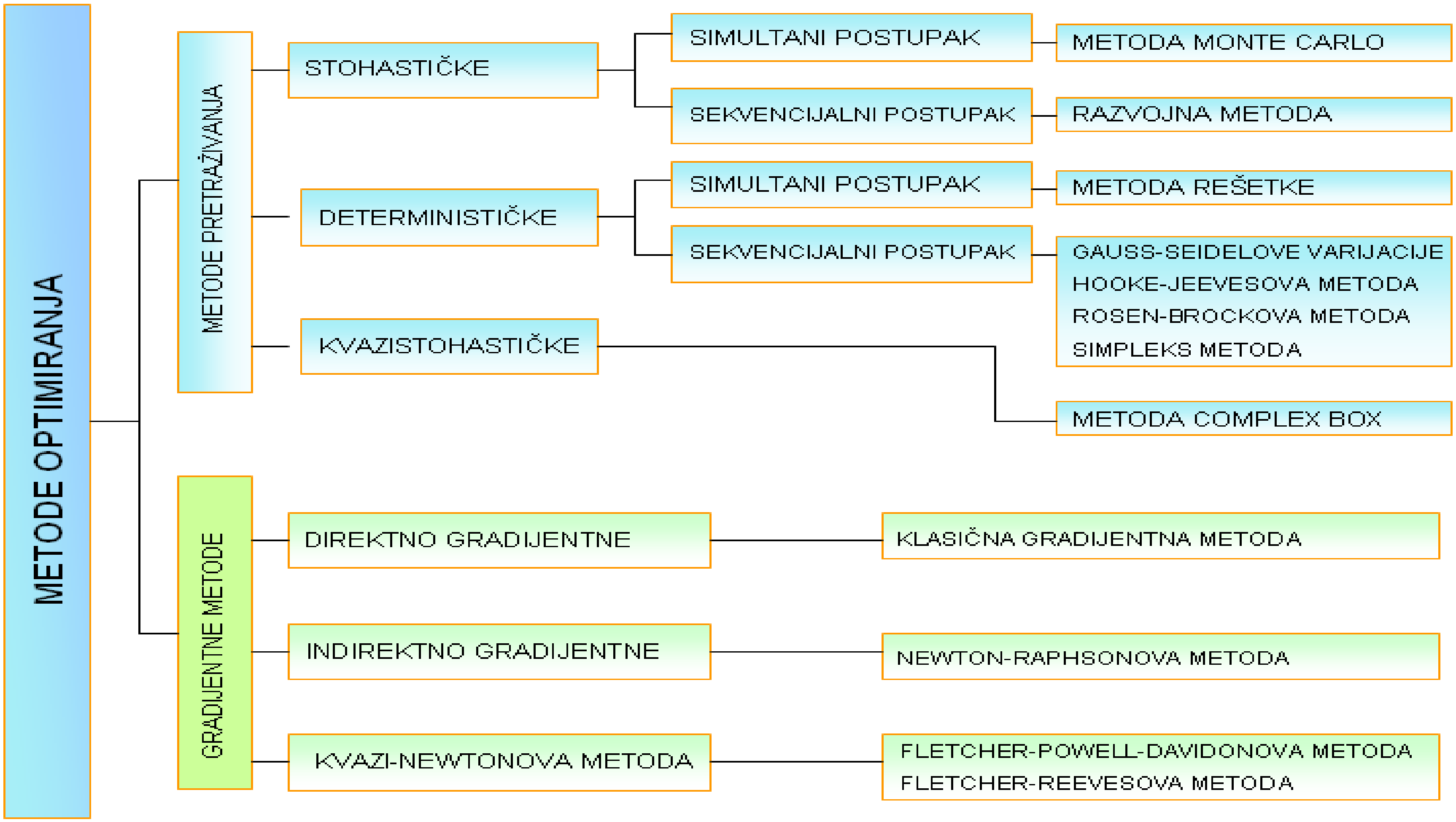
Metode pretraživanja

Općenito metode pretraživanja kao što im i samo ima govori pretražuju prostor mogućih rješenja poštujući određena pravila pretrage odnosno algoritme prema čemu se međusobno i razlikuju. Pretraživački algoritmi za nelinearne probleme koriste linijsku minimalizaciju za rješavanje potproblema koji na koncu vode do određivanja optimalne vrijednosti funkcije cilja *f*(*X*). Algoritam pretražuje uzduž jednog vetora dok ne pronađe minimalnu vrijednost funkcije cilja na tom vektoru. Nakon toga pretraga se nastavlja uzduž drugog vektora koji je ortogonalan na prvi vektor. Linijska pretraga nastavlja se uzduž novog vektora dok ne pronađe i njegovo minimalno rješenje. Proces linijske minimalizacije nastavlja se kontinuirano sve dok pretraživački algoritam ne pronađe optimalno rješenje.

Gradijentne metode

Gradijentne metode su determinističke sekvencijalne metode koje se koriste gradijentom funkcije cilja kako bi utvrdile smjer najbržeg porasta ili opadanja funkcije cilja. Kada se utvrde točke u kojima je gradijent jednak nuli ispituje se vrijednost druge derivacije u tim točkama kako bi se odredila priroda ekstrema.

Gradijentne, determinističke i kvazistohastične metode zajedno čine skupinu «metoda penjanja uzbrdo». Simultani i sekvencijalni postupci razlikuju se prema teoretskom i praktičnom utrošku programiranja.Pod teoretskim utroškom programiranja smatra se izrada algoritma u kojem je upotrebljena minimalna količina operacija nad podacima koje dovode do optimalnog rješenja problema. Općenito, prilikom praktične realizacije teoretski vrlo dobro razrađenih sustava javljaju se brojna ograničenja nametnuta zbog nesavršenosti ljudi i tehnologije koja uzrokuju slabije performanse stvarnih sustava od teoretski mogućih performansi. Iz istog razloga stvarni algoritmi pisani su često na način da provode veći broj operacija nad podacima nego što je to teoretski potrebno. To uzrokuje povećanje potrebnog vremena za vršenje proračuna što posebno dolazi do izražaja prilikom računanja kompleksnih problema sa više varijabli i složenim funkcijama ograničenja i funkcijom cilja.
Na slici 2.2. prema prikazana je sistematizacija metoda NP .



Slika 2.2. Sistematizacija metoda nelinearnog statičkog programiranja

2.3. Podjela metoda optimiranja prema broju varijabli

Metode nelinearnog programiranja se prema broju varijabli dijele na:

- metode za funkcije jedne varijable
- metode za funkcije više varijabli

Tablica 2.1. Podjela metoda nelinearnog programiranja prema broju varijabli

	Funkcije jedne varijable	Funkcije više varijabli
metode koje koriste derivacije	▪ Newtonova metoda	▪ Metoda eliminacije ▪ Metoda Lagrangeovih multiplikatora ▪ Metoda kaznenih funkcija ▪ Metoda najbržeg uspona ▪ Metoda ograničenog gradijenta ▪ Metoda konveksnih kombinacija ▪ Newton-Raphsonova metoda ▪ Kvazi-Newtonova metoda ▪ Fletcher-Powell-Davidonova metoda ▪ Fletcher-Reevesova metoda
metode koje ne koriste derivacije	▪ Fibonacijeva metoda ▪ Metoda zlatnog reza ▪ Metoda aproksimacije polinomom ▪ Metoda uniformnog traženja ▪ Metoda račvanja	▪ Metoda Monte Carlo ▪ Rehnbergova razvojna strategija ▪ Metoda rešetke ▪ Metoda Gauss-Seidelove varijacije ▪ Nelder-Meadova metoda ▪ Metoda complex-box ▪ Hooke-Jeevesova metoda ▪ Powellova metoda ▪ Rosenbrockova metoda

3. Nelinearne funkcije cilja u tehnološkim procesima

3.1.Optimizacija tehnoloških parametara tokarenja za čelik 40CrMnMo

Iz literature [11] pribavljeni su svi potrebni podaci.

- **Podaci o materijalu**

Čelik 40CrMnMo ima vrlo široku primjenu u brojnim industrijama kao npr. u automobilskoj, prehrambenoj i kozmetičkoj industriji stoga je s ekonomskog aspekta sasvim opravdano optimirati tehnološke parametre obrade ovog čelika.

- **Modeliranje**

Kada se koriste veće brzine rezanja zona rezanja se zagrijava još intenzivnije tako da se materijal zagrijava što pozitivno utječe na životni vjek alata, no ako su brzine rezanja prevelike postoji opasnost od spalljivanja granica zrna i pojavljivanja pukotina tako da je potrebno optimirati parametre na taj način da se ne uzimaju u obzir one vrijednosti parametara koje rezultiraju lošom kvalitetom obrade.
Kako bi se utvrdile granice parametara obrade potrebno je bilo provesti početna ispitivanja . Nakon provedbe početnih ispitivanja autori [ ] su utvrdili da alata pri većim brzinama rezanja zbog povećanog trošenja alata brzina rezanja ne bi smjela prelaziti 650 m/min te da ne bi smjela biti manja od 450 m/min. Zbog fizičkih svojstava materijala minimalni posmak iznosi f_{min} = 0.1 mm dok je maksimalni posmak f_{max} = 0.2 mm te minimalna dubina rezanja a_{pmin} = 0.2 mm i maksimalna dubina rezanja a_{pmax} = 0.35 mm.
Nakon utvrđivanja granica proveden je plan pokusa sa 32 mjerenja (centralni kompozitni plan mjerenja 2^4=32 sa 8 mjerenja u centru). Svi pokusi provedeni su bez upotrebe sredstva za hlađenje.

Kao rezultat pokusa autori su došli do sljedećih izraza:

$$R_a = 4{,}829001 - 0{,}00632 \cdot v_c - 7{,}280413 \cdot a_p + 1{,}554342 \cdot f -$$
$$-3{,}567335 \cdot r_e + 4{,}314476 \cdot 10\text{-}6 \cdot v_c^2 + 15.833074 \cdot a_p^2 +$$
$$+20{,}124416 \cdot f^2 + 1{,}776459 \cdot r_e^2$$

- **Postavljanje matematičkog modela**

Potrebno je minimizirati prosječno odstupanje profila Ra :

$$R_a = Fc(v_c, a_p, f, r_e) = 4{,}829001 - 0{,}00632 \cdot v_c - 7{,}280413 \cdot a_p + 1{,}554342 \cdot f -$$
$$-3{,}567335 \cdot r_e + 4{,}314476 \cdot 10\text{-}6 \cdot v_c^2 + 15.833074 \cdot a_p^2 +$$
$$+20{,}124416 \cdot f^2 + 1{,}776459 \cdot r_e^2 \rightarrow \text{MIN} \ [\mu m]$$

Podložno ograničenjima:

(1) $375 \le v_c \le 675$ [m/min]

(2) $0{,}125 \le a_p \le 0{,}35$ [mm]

(3) $0{,}05 \le f \le 0{,}2$ [mm]

(4) $0{,}4 \le r_e \le 1{,}2$ [mm]

- **Rješavanje problema**

Problem je rješen metodom nasumične pretrage. Generirano je 20 000 slučajnih brojeva za svaku varijablu.

U tablici 3.1. prikazano je pravilo po kojem se dodjeljuju vrijednosti varijabli u ovisnosti o slučajnom broju za svaku od 20 000 iteracija.

Vrijednost 0 je dodjeljena minimalnom dozvoljenom iznosu varijable dok je vrijednost 0,9 dodjeljena maksimalnom iznosu varijable.

Korak se računao na način da se dozvoljeni raspon varijabli podjelio na deset jednakih dijelova npr. za brzinu: $k_{var} = R_{var}/10$ odnosno $k_{va\,r} = (675\text{-}375)/10$

Tablica 3.1. Generiranje vrijednosti varijabli ovisno o vrijednosti sl. broja

lookup	v_c	a_p	f	r_e
0	375	0,125	0,05	0,4
0,1	405	0,1475	0,065	0,48
0,2	435	0,17	0,08	0,56
0,3	465	0,1925	0,095	0,64
0,4	495	0,215	0,11	0,72
0,5	525	0,2375	0,125	0,8
0,6	555	0,26	0,14	0,88
0,7	585	0,2825	0,155	0,96
0,8	615	0,305	0,17	1,04
0,9	645	0,3275	0,185	1,12
1	675	0,35	0,2	1,2

Nakon proračuna funkcije cilja za svih 20 000 iteracija rješenje iznosi:

R_a (µm)	v_c (m/min)	a_p (mm)	f (mm)	r_e (mm)	Iteracija
0,0509349	645	0,2	0,05	1,04	9407

Sljedeći korak je pretraga u okolici dobivene prve točke (645 , 0,2 , 0,05 , 1,04) na način da vrijednosti varijable ovise o slučajnim brojevima kao što je to prikazano na tablici 3.2.

Tablica 3.2. Pravilo novog generiranja vrijednosti varijabli

lookup	v_c	a_p	f	r_e
0	620	0,125	0,05	0,89
0,1	625	0,1475	0,065	0,92
0,2	630	0,17	0,08	0,95
0,3	635	0,1925	0,095	0,98
0,4	640	0,215	0,11	1,01
0,5	645	0,2375	0,125	1,04
0,6	650	0,26	0,14	1,07
0,7	655	0,2825	0,155	1,1
0,8	660	0,305	0,17	1,13
0,9	665	0,3275	0,185	1,16
1	670	0,35	0,2	1,19

Ponovnim provođenjem postupka dobiveno je sljedeće rješenje:

R_a (µm)	v_c (m/min)	a_p (mm)	f (mm)	r_e (mm)	Iteracija
0.035342	665	0,2	0,05	1,01	16214

Može se zaključiti da je dobiveno rješenje bolje od prethodnog ali nije idealno iz razloga što je relativno velik korak k=15 između vrijednosti brzine u tablici 3.2. tako da je potrebno ponoviti postupak ponovno kako bi se točnije odredila brzina rezanja.
Budući da ovaj problem ima četiri varijable postoji opasnost kada bi se generiralo varijable koristeći manji korak da nebi bilo dovoljan broj iteracija da se pogodi kombinacija koja sadrži optimalne iznose varijabli. Iz tog razloga provedena su tri ponavljanja proračuna i to tako da se za zadnji proračun varijable generiraju prema tablici 3.3.

Tablica 3.3. Pravila posljednjeg generiranja vrijednosti varijabli

lookup	v_c	a_p	f	r_e
0	655	0,15	0,05	0,88
0,1	657	0,16	0,065	0,9
0,2	659	0,17	0,08	0,92
0,3	661	0,18	0,095	0,94
0,4	663	0,19	0,11	0,96
0,5	665	0,2	0,125	0,98
0,6	667	0,21	0,14	1
0,7	669	0,22	0,155	1,02
0,8	671	0,23	0,17	1,04
0,9	673	0,24	0,185	1,06
1	675	0,25	0,2	1,08

Usvaja se dobiveno rješenje:

R_a (µm)	v_c (m/min)	a_p (mm)	f (mm)	r_e (mm)	Iteracija
0,0300191	673	0,2	0,05	1,01	12523

- **Zaključak**

S obzirom da optimalno rješenje problema iznosi: R_a=0,0290115µm , v_c=675m/min, a_p= 0,2375mm, r_e=1 može se zaključiti da metoda nasumične pretrage daje zadovoljavajući rezultat uz minimalna odstupanja.

3.2.Izrada podloga za odlučivanje za tehnološku pripremu rada za tokarenje čelika 40CrMnMo

U slučaju primjene više različitih tokarskih noževa može se dogoditi da oni nemaju jednak radijus zaobljenja r_e stoga je korisno izraditi tablice u kojima bi se optimirali parametri brzine rezanja, posmaka i dubine rezanja ovisno o radijusu zaobljenja noža odnosno o izabranom nožu.

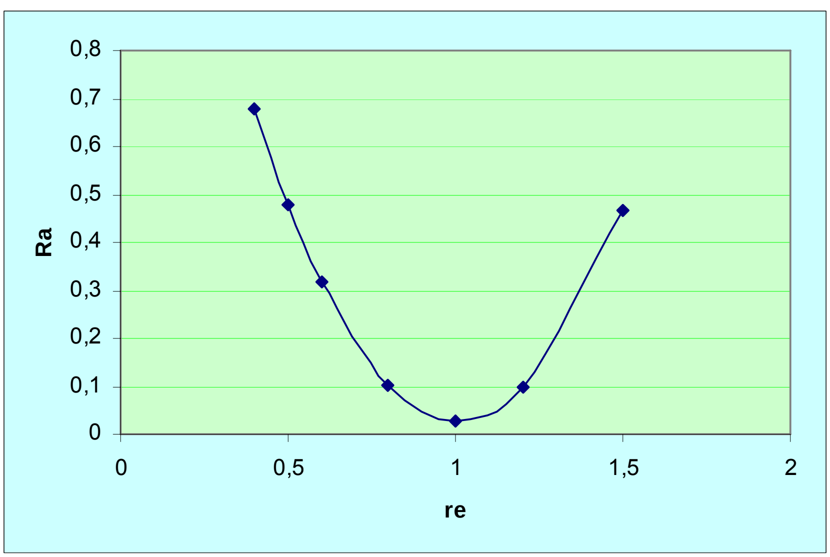
Kada radnik mjenja nož tada jednostavno može isčitati iz tablice koji parametre treba postaviti kako bi se postigla minimalna hrapavost površine ili neka druga veličina koju optimiramo ovisno o radijusu zaobljenja izabranog novog noža.

U tablici 3.4. prikazan je optimalan iznos parametara za nekoliko različitih iznosa r_e.

Tablica 3.4. Ovisnost funkcije cilja o radijusu alata uz optimalne parametre

r_e	V (m/min)	a_p (mm)	F (mm)	R_a (µm)
0,4	675	0,2295	0,05	0,677189
0,5	675	0,2295	0,05	0,480337
0,6	675	0,2295	0,05	0,319014
0,8	675	0,23	0,05	0,102953
1	675	0,232	0,05	0,02908
1,2	675	0,23	0,05	0,097186
1,5	675	0,23	0,05	0,465917

Na slici 3.1. grafički su prikazani podaci iz tablice 3.4.



Slika 3.1. Ovisnost funkcije cilja o radijusu alata uz optimalne parametre

Iz slike se vidi da se čak i uz optimalne iznose ostalih parametara vrijednost R_a bitno ovisi o veličini r_e te da je najmanja vrijednost R_a upravo kada r_e iznosi 1mm tako da se preporuča koristiti alate sa tim radijusom zaobljenja kada je god to moguće.

3.3.Optimizacija tehnoloških parametara glodanja

Funkcija cilja preuzeta je iz literature [9], Šprljan M., *Diplomski rad*, Fakultet strojarstva i brodogradnje sveučilišta u Zagrebu, Zagreb 2010. gdje je potrebno optimizirati prosječno odstupanje profila Ra za tehnološki proces glodanja. Varijable odnosno parametri koji utječu na funkciju cilja su brzina rezanja v, posmak f i kut namještanja alata φ.

Radi se o linearnoj funkciji cilja sa linearnim ograničenjima stoga se ovaj problem mogao riješiti nekom od metoda za nelinearno programiranje, no u sljedećem primjeru problem je riješen metodom nasumične pretrage kako bi se pokazalo da je ta metoda upotrebljiva i za rješavanje linearnih problema.

- **Matematički model**

Potrebno je minimizirati funkciju:

R_a=0,237-0,00175·v+8.693·f-0,00159·φ→MIN [μm]

Podložnu ograničenjima:

(1) 124,53 ≤ v ≤ 167,03 [m/min]

(2) 0,025 ≤ f ≤ 0,083 [mm]

(3) 6,2° ≤ φ ≤ 14,8 °

- **Rješavanje problema**

Problem je rješen metodom nasumične pretrage. Generirano je 5 000 slučajnih brojeva za svaku varijablu. U tablici 3.5. prikazano je dodjeljivanje vrijednosti varijabli ovisno o iznosima slučajnog broja.

Tablica 3.5. Pravilo prvog generiranja varijabli (glodanje)

lookup	v	f	φ
0	124,53	0,025	6,2
0,1	128,78	0,0308	7,06
0,2	133,03	0,0366	7,92
0,3	137,28	0,0424	8,78
0,4	141,53	0,0482	9,64
0,5	145,78	0,054	10,5
0,6	150,03	0,0598	11,36
0,7	154,28	0,0656	12,22
0,8	158,53	0,0714	13,08
0,9	162,78	0,0772	13,94
1	167,03	0,083	14,8

Dobiveni početni rezultat iznosi:

Ra (μm)	v_c (m/min)	f(mm)	φ (°)	Iteracija
0,147295	162,78	0,025	13,94	558

Provodi se sljedeći proračun u kojem se varijable generiraju prema pravilu prikazanom u tablici 3.6. gdje se pretražuje područje oko brzine 162,78 m/min i oko kuta 13,94° dok se pretražuju posmaci veći od 0,025 mm/okr jer zbog ograničenja manji posmaci 0,025 ne dolaze u obzir.

Tablica 3.6. Pravilo drugog generiranja varijabli (glodanje)

lookup	v	f	φ
0	162.68	0.025	13.89
0.1	162.7	0.0255	13.9
0.2	162.72	0.026	13.91
0.3	162.74	0.0265	13.92
0.4	162.76	0.027	13.93
0.5	162.78	0.0275	13.94
0.6	162.8	0.028	13.95
0.7	162.82	0.0285	13.96
0.8	162.84	0.029	13.97
0.9	162.86	0.0295	13.98
1	162.88	0.03	13.99

Nakon drugog generiranja varijabli dobiven je rezultat :

Ra (µm)	v_c(m/min)	f(mm)	φ (°)	Iteracija
0,147092	162,86	0,025	13,98	1683

Problem je riješen pomoću rješavaća „Solver“ u programskom paketu Excel pri čemu je dobiveno stvarno optimalno rješenje:

Ra (µm)	v_c (m/min)	f (mm)	φ (°)
0,138491	167,03	0,025	14,8

Usvaja se drugi rezultat kao dovoljno dobra aproksimacija optimalnog rješenja.
Ukoliko se želi postići bolje rješenje potrebno je ponavljati postupak dok se ne postigne zadovoljavajuća točnost.

- **Zaključak**

Metodu nasumične pretrage može se upotrijebiti za rješavanje linearnih funkcija cilja tako da je moguće programirati univerzalne algoritme koji se mogu koristiti za rješavanje NP i LP problema.

3.4.Optimizacija tehnoloških parametara brušenja

Funkcija cilja je preuzeta iz literature [10] gdje je potrebno optimizirati prosječno odstupanje profila *Ra* za tehnološki proces brušenja.
Varijable odnosno parametri koji utječu na funkciju cilja su brzina rezanja *v* (odnosno x_1) i posmak *f* (odnosno x_2).Parametri su zadani u transformiranom obliku tako da vrijednost -1 odgovara minimalnom dozvoljenom iznosu parametra a vrijednost 1 maksimalnom, pri čemu nula označava sredinu dozvoljenog raspona. Zbog upotrebe transformiranih parametara i funkcija cilja je transformirana tako da je ona fizički gledano bezdimenzijska veličina koju se može transformirati u odstupanje profila izraženo u [µm]. S aspekta rješavanja problema to nije bitno, važno je samo točno izračunati transformirane parametre.

- **Matematički model**

Potrebno je minimizirati funkciju:

$f(X)=232{,}37 +15{,}677\cdot x_1-65{,}495\cdot x_2+x_1\cdot(x_2-29{,}25)+x_1\cdot(x_1-39{,}196)+x_2\cdot(x_2-21{,}779)$

Podložnu ograničenjima:

(1)$1\geq x_1\geq -1$

(2)$1\geq x_2\geq -1$

- **Rješavanje problema**

Problem je riješen metodom nasumične pretrage.

Tablica 3.7. Pravilo prvog generiranja varijabli (brušenje)

lookup	X_1	X_2
0	-1	-1
⋮	⋮	⋮
0,465	-0,07	-0,07
0,47	-0,06	-0,06
0,475	-0,05	-0,05
0,48	-0,04	-0,04
0,485	-0,03	-0,03
0,49	-0,02	-0,02
0,495	-0,01	-0,01
0,5	0	0
0,505	0,01	0,01
0,51	0,02	0,02
0,515	0,03	0,03
0,52	0,04	0,04
⋮	⋮	⋮
1	1	1
0,545	0,09	0,09

U tablici 3.7. prikazano je pravilo po kojem se dodjeljuju vrijednosti varijablama. Moguće je generirati 200 različitih vrijednosti varijabli pri čemu se generiralo 5000 vrijednosti za svaku varijablu.

Dobiveno je sljedeće rješenje:

$F(X)$	x_1	x_2	Iteracija
95,327	1	1	81

Dobiveno rješenje je ujedno i optimalno rješenje.

4. Programski paketi pogodni za pronalaženje optimuma nelinearnih funkcija

4.1.Podjela prema funkcionalnosti

U zadnja dva desetljeća došlo je do značajnog usavršavanja optimizacijskog software-a što je učinilo mogućim rješavanje brojnih problema u područjima znanosti i inženjerstva. Usavršenja optimizacijskih algoritama i računalne tehnologije omogućila su rješavanje optimizacijskih problema sa nekoliko tisuća do nekoliko milijuna ograničenja i varijabli.

Prema funkcionalnosti optimizacijski software se može poijeliti u pet skupina.

U prvu skupinu spada software za općenitu znanstvenu obradu i analizu podataka . Primjer ove skupine softwera je Excell, Mathematica, MATLAB i SAS.

Ti software-ski sustavi uključuju optimizacijske module kao i mnoge druge računske i grafičke alate te nude širok spektar mogućnosti, premda često ne uključuju napredne alate za rješavanje velikih ili posebno izazovnih optimizacijskih problema.

Kako ti sustavi nisu specijalizirani za optimizaciju nerjetko ne raspolažu sa optimizacijskim alatima koji olakšavaju razvoj i implementaciju optimizacijskih modela.

U drugu skupinu spada software za modeliranje posebno usmjeren na optimizaciju. Korisniku je omogućeno olakšano formuliranje optimizacijskih problema u odgovarajući model i primjena velikog broja optimizacijskih alata za njihovo rješavanje. Jezik za modeliranje tada transformira model u matematički oblik (formu) koji software zahtijeva. Ukoliko se radi o nelinearnim modelima software može iznaći formule za izvođenje derivabilnih operacija. Jezici za modeliranje omogućuju stvaranje optimizacijskih modela koji se zasnivaju na setovima upisanih podataka tako da oni sami vrše parametrizaciju. To je naročito pogodno u primjenama kad je generalni model postavljen ali je prisutna redovita promjena podataka.

Ako dođe i do male promjene na modelu rješavanje zahtjeva drugačiji solver (rješavač), na primjer ako se doda nelinearni izraz (kao ograničenje ili u funkciju cilja) nekom linearnom modelu software automatski izabire optimalan alat iz svog arsenala alata za rješavanje novopostavljenog problema.

Treća kategorija software-a su rješavaći za linearno programiranje kao i software za rješavanje problema cijelobrojnog programiranja. Postoji nekoliko razloga zbog kojih se software za linearno programiranje izdvaja iz ostalih općenitijih optimizacijskih algoritama. Jedan od razloga je to što je lakše opisati model jer je potrebno jedino specificirati vektore i matrice a ne cjelokupne nelinearne funkcije.

Drugi razlog je to što software za linearno programiranje sadrži i dodatke koje ne sadrže ostali software-i za nelinearnu optimizaciju kao što su podešavanje osjetljivosti (preciznosti) analize te mogučnost određivanja cijelobrojnih iznosa varijabli, iznimka su neki software-i za kvadratno programiranje.

Četvrta kategorija su rješavači za kvadratno programiranje. Kvadratni modeli su nelinearni ali oni mogu kao i linearni programi biti opisani specificirajučim vektorima i matricama što je bolji pristup od opisivanja nelinearnim funkcijama. Konveksni kvadratni problemi mogu se uspješno rješiti koristeći metodu unutarnje točke ili modificiranu simpleks metodu. To je razlog zbog kojeg su najčešće rješavači za linearno i kvadratno programiranje često kombinirani u jednom software-skom paketu koji sadrži korisne značajke kao što je mogućnost računanja sa cijelobrojnim varijablama ili podešavanje preciznosti analize. Kako se nelinearne optimizacijske metode često zasnivaju na kvadratnim modelima, te često uzimaju u obzir nelinearna ograničenja odvojeno od nelinearnih software za nelinearnu optimizaciju bi trebao dobro rješavati konveksne kvadratne programe također.

U slučaju kada treba rješiti nekonveksne kvadratne probleme ili probleme sa kvadratnim ograničenjima koji imaju lokalna rješenja (lokalne optimume) tada je nužno koristiti rješavače za općenitu (generalnu) nelinearnu optimizaciju.

Rješavači za općenitu ili osnovnu nelinearnu optimizaciju su peta kategorija optimizacijskih software-a. Postoje brojne varijante modela za nelinearnu optimizaciju sa pripadajućim arsenalom algoritama za nelinearnu optimizaciju. Ako neki problem predstavlja poseban izazov za rješavanje, korisnik bi trebao pribaviti dodatni vodić za odabir rješavača kako bi odabrao prikladan rješavač ili čak reformulirao model kako bi ga se moglo rješiti dostupnim metodama. Ponekad je teško izraziti nelinearni optimizacijski model u software-u tako da se mora izraziti pod uvjetima drugog pomoćnog software-a.

Novi software-i za linearno programiranje mogu rješiti probleme sa milijunima varijabli i ograničenja pri čemu uvijek nalaze globalni optimum za rješenje, no ako je postavljen uvjet da neke ili sve varijable moraju biti cjelobrojne teže je doći do rješenja što znatno ograničava raspon problema koji mogu biti rješeni.

U tim slučajevima postojeće verzije rješavača često nisu sposobne odrediti globalno rješenje već nam nude riješenje koje odudara u postocima od globalnog rješenja odnosno koje se nalazi u određenim tolerancijskim granicama od optimuma.

Drugim rječima usvaja se rješenje za koje proizvođeć software-a tvrdi da se razlikuje od globalnog optimuma za ne više od npr. 5%.

Nelinearni problemi imaju često više lokalnih optimuma tako da većina software-a za nelinearno optimiranje garantira da je pronašla jedno od lokalnih rješenja dok ne može jamčiti da je to ujedno globalni optimum (stoga se uspoređuju rješenja raznih software-a za rješavanje istog problema kako bi se utvrdilo koji je od njih bolji za zadani problem). Ako su funkcije ograničenja nelinearne često software ne može odrediti niti *pogodnu točku (eng. Feasible point)* , u tim slučajevima jako je korisno pogoditi u kojem se području nalazi globalni ekstrem koje se naziva njegovim „susjedstvom“ u stranoj literaturi.

Susjedstvo globalnog rješenja je područje koje je samo mali dio ukupnog n-dimenzionalnog prostora mogućih rješenja (gdje je n broj varijabli).

U nekim slučajevima potrebno je koristiti više različitih pogađanja područja (susjedstva) kako bi se izbjeglo usvajanje lokalnog optimuma za globalni. Algoritmi rade na principu da generiraju onoliko rješenja koliko je bilo područja pretrage nakon čega se određuju rješenja za svako područje. Dobivena se rješenja uspoređuju nakon čega se izabire najmanje ili najveće (ovisno traži li se minimum ili maksimum) za konačno rješenje

U tablici 4.1. Opisane su metode koje se praktično koriste u programskim paketima za provođenje matematičkih operacija iteracije, interpolacije, aproksimacije i rješavanja sustava jednadžbi.

Tablica 4.1.Osnove numeričke matematike korištene u programskim paketima

		METODE
POSTUAK	ITERACIJE	*1.Metoda krivog polozaja* *2.Kvadratna metoda* *3.Opca iteracija* *4.Newton-Raphsonova metoda*
	INTERPOLACIJE	*1.Lagrangeova interpolacija* *2.Newtonova interpolacija* *3.Interpolacija s razlomljenim eksponentima polinoma* *4.Interpolacija elememtarnim funkcijama* *5.Spline interpolacija*
	APROKSIMACIJE	*1.Aproksimacije polinomom kod metode najmanjih kvadrata za probleme sa ograničenjima i bez njih* *2.Aproksimacija eksponencijalnim i logaritamskim funkcijama* *3.Ortogonalni polinomi* *4.Aproksimacija trigonomerijskim funkcijama,* *5.Aproksimacija hiperbolicnim funkcijama s primjenom u mehanici* *6.Spline aproksimacije*
	RJEŠAVANJE SUSTAVA JEDNADŽBI	*1.Pravila rješavanja sustava linearnih jednadzbi* *2.Metode za rješavanje sustava nelinearnih jednadzbi :* • *Newton - Raphsonova metoda* • *Metoda opce iteracije* *3.Rjesavanje diferencijalnih jednadzbi*

Popis programskih paketa pogodnih za NLP:

- LINGO
- Evolver
- Maple
- Matlab
- Mathematica
- S-PLUS
- LabVIEW
- Scilab
- Free Mat
- GNU Octave
- AIMMS

Vrlo često korišteni software-ski paket su Matlab i Mathematica stoga su dolje navedeni dodatci (solveri) za nadogradnju koji sadrže alate za rješavanje različitih modela nelinearnog programiranja .

Nadogradnja (rješavači) za matlab:

1. *Optimization Toolbox* (Sadrži rutine za rješavanje brojnih NLP problema sa ograničenjima i bez ograničenja)
2. *TOMLAB Optimization environment*
3. *LOQO* (Pruža izvrsno sučelje za kvadratno programiranje)
4. *MCS* (Služi za određivanje globalnih ekstrema)

Nadogradnja (rješavači) za Mathematicu:

1. *MathOptimizer* (Za rješavanje velikog broja generalnih NLP problema ,razvijen posebno za guste i visoko nelinearne sustave u kojima postoji veliki i nepoznati broj lokalnih optimuma)
2. *Global Optimization* (Veliki skup funkcija koje se koriste za određivanje globalnog ekstrema)

4.2.Opis nekih programskih paketa pogodnih za NP

4.2.1.Lingo

Lingo je alat za lakše, brže i učinkovitije postavljanje i rješavanje linearnih i nelinearnih problema.

Ključne prednosti koje nudi LINGO su:

1. Jednostavno izražavanje modela
   (LINGO omogućava brzu formulaciju linearnih, nelinearnih i cjelobrojnih problema . Jednom formulirani problemi prikazani su kao da su napisani na papiru što olakšava rad i skraćuje vrijeme adaptacije korisnika.)

2. Napredno upravljanje podacima
   (Moguće je formirati modele koji izvlače i koriste podatke iz baza podataka ili iz radnih listova drugih kompatibilnih programa. Nakon optimiranja moguće je postići da rezultati budu spremljeni u istu ili neku drugu bazu podataka ili radni list .)

3. Ugrađeni su snažni rješavači (solveri)
   (LINGO uključuje veliki skup rješavača za linearno i nelinearno programiranje. Uspješno rješava probleme stohastičke i cjelobrojne optimizacije kao i probleme kvadratnog programiranja, probleme s kvadratnim ograničenjima te ostale nelinearne probleme.

4. Nije potrebna ugradnja drugih rješavača
   (Na temelju upisane formulacije problema Lingo automatski izabire jedan od svojih rješavača koji daje najbolje rezultate za tu vrstu problema.)

4.2.2.Evolver

Evolver je rješavač za Microsoft Excell koji služi za optimizaciju linearnih i nelinearnih sustava. U Evolveru je uspješno primjenjena inovativna tehnologija genetičkih algoritama koja u vrlo kratkom vremenu rješiava optimizacijske probleme u financijama, distribuciji, planiranju, raspodjeli resursa, proizvodnji i drugim srodnim područjima. Načelno svaki problem koji je moguće postaviti u Excellu riješiv je sa Evolverom.
Genetički algoritmi Evolvera brzo i točno dolaze do globalnog optimuma što nije bio slučaj za tradicionalnije verzije dodataka za Excell.

4.2.3.GNU octave

GNU Octave je besplatan software koji se može redistribuirati i usavršavati u svrhu promicanja znanja. Nakon usavršavanja omogučava riješavanje složenijih problema koje osnovna verzija nije u stanju riješiti. Uspješno se može primjeniti za pronalaženje kritičnih točaka nelinearnih jednadžbi. Kompatibilan je sa Matlab-om kao i Scilab koji je također besplatan.

4.2.4.AIMMS

U AIMMS-u je uspješno primjenjen proračun pomoću Hesseovih matrica koji može ubrzati traženje riješenja kod nekih NLP rješavača. Kako uspješnost raznih NLP rješavača ovisi o specifičnom problemu, AIMMS veoma brzo pronalazi najbolji rješavač za zadni problem jer podržava vezu sa velikim brojem dostupnih rješavača.

5. Zaključak

Razvoj raznih metoda pomoću kojih se mogu modelirati funkcije cilja uzrokovao je veću primjenu nelinearnog programiranja u praksi.

Rezultat plana pokusa često su funkcije cilja drugog i trećeg reda sa mnogo članova tako da u nekim slučajevima postoji veći broj lokalnih optimuma stoga je potrebno izabrati prikladnu metodu za rješavanje zadanog problema.

Od brojnih dostupnih metoda za NP uz genetske algoritme ističe se metoda nasumične pretrage kao vrlo korisna metoda kojom se mogu rješavati nelinearni problemi u kojima je funkcija cilja diskontinuiranog oblika, nederivabilna, prekompleksna za izvođenje derivacija ili koja ima veliki broj lokalnih optimuma što joj daje prednost pred velikim brojem klasičnih metoda.

Uz to navedena metoda lako se programira te je vrlo korisna za detekciju u kojem se području nalazi globalni ekstrem. U nekim slučajevima moguće je koristiti neku točniju metodu nakon što je detektirano područje globalnog ekstrema premda metoda nasumične pretrage može također nakon određenog broja pretraga dovesti do rezultata zadovoljavajuće točnosti.

Ubrzani razvoj suvremenih numeričkih metoda za rješavanje nelinearnih problema i računalne opreme uzrokovati će veću upotrebu nelinearnog programiranja kao područja koje je dugo vremena bilo zapostavljano zbog ljudskih i tehničkih ograničenja .

6. Literatura

[1] Singiresu, S. Rao. , *Engineering Optimization - theory and practice*, John Wiley & Sons, Inc., New Jersey, 2009.

[2] www.wiki.mcs.anl.gov

[3] Šakić N., Štefanić N. , *Metode optimiranja proizvodnje, Inženjerski priručnik,* Fakultet strojarstva i brodogradnje sveučilišta u Zagrebu, Zagreb

[4] Kalpić, D., Mornar, V.,*Operacijska istraživanja*, ZEUS, Zagreb, 1996.

[5] Griva, I., Stephen G. Nash, Sofer, A. ,*Linear and nonlinear optimization*, Society for Industrial and Applied Mathematics, Philadelphia, 2009.

[6] Barković, D.,*Operacijska istraživanja*, Ekonomski fakultet, Osijek, 2002.

[7] www.fsb.hr/ship-design/optim.htm

[8] www.gnu.org

[9] Šprljan M., *Diplomski rad*, Fakultet strojarstva i brodogradnje sveučilišta u Zagrebu, Zagreb 2010.

[10] Rajić D., *Diplomski rad*, Fakultet strojarstva i brodogradnje sveučilišta u Zagrebu, Zagreb 2010.

[11] Stoić A.,Kopač J.,Cukor G.*Testing of machinability of mould steel 40CrMnMo7 using genetic algorithm*, www.sciencedirect.mht